# Approximating Likelihood Ratios with Calibrated Discriminative Classifiers

Kyle Cranmer[1], Juan Pavez[2] and Gilles Louppe[1]
[1]New York University
[2]Federico Santa María University

March 14, 2016

### Abstract

In many fields of science, generalized likelihood ratio tests are established tools for statistical inference. At the same time, it has become increasingly common that a simulator (or generative model) is used to describe complex processes that tie parameters $\theta$ of an underlying theory and measurement apparatus to high-dimensional observations $\mathbf{x} \in \mathbb{R}^p$. However, simulator often do not provide a way to evaluate the likelihood function for a given observation $\mathbf{x}$, which motivates a new class of likelihood-free inference algorithms. In this paper, we show that likelihood ratios are invariant under a specific class of dimensionality reduction maps $\mathbb{R}^p \mapsto \mathbb{R}$. As a direct consequence, we show that discriminative classifiers can be used to approximate the generalized likelihood ratio statistic when only a generative model for the data is available. This leads to a new machine learning-based approach to likelihood-free inference that is complementary to Approximate Bayesian Computation, and which does not require a prior on the model parameters. Experimental results on artificial problems with known exact likelihoods illustrate the potential of the proposed method.

*Keywords:* likelihood ratio, likelihood-free inference, classification, particle physics

# 1 Introduction

The likelihood function is the central object that summarizes the information from an experiment needed for inference of model parameters. It is key to many areas of science that report the results of classical hypothesis tests or confidence intervals using the (generalized or profile) likelihood ratio as a test statistic. At the same time, with the advance of computing technology, it has become increasingly common that a simulator (or generative model) is used to describe complex processes that tie parameters $\theta$ of an underlying theory and measurement apparatus to high-dimensional observations $\mathbf{x}$. However, directly evaluating the likelihood function in these cases is often impossible or is computationally impractical.

The main result of this paper is to show that the likelihood ratio is invariant under dimensionality reductions $\mathbb{R}^p \mapsto \mathbb{R}$, under the assumption that the corresponding transformation is itself monotonic with the likelihood ratio. As a direct consequence, we derive and propose an alternative machine learning-based approach for likelihood-free inference that can also be used in a classical (frequentist) setting where a prior over the model parameters is not available. More specifically, we demonstrate that discriminative classifiers can be used to construct equivalent generalized likelihood ratio test statistics when only a generative model for the data is available for training and calibration.

As a concrete example, let us consider searches for new particles at the Large Hadron Collider (LHC). The simulator that is sampling from $p(\mathbf{x}|\theta)$ is based on quantum field theory, a detailed simulation of the particle detector, and data processing algorithms that transform raw sensor data into the feature vector $\mathbf{x}$ (Sjostrand et al., 2006; Agostinelli et al., 2003). The ATLAS and CMS experiments have published hundreds of papers where the final result was formulated as a hypothesis test or confidence interval using a generalized likelihood ratio test (Cowan et al., 2010), including most notably the discovery of the

Higgs boson (The ATLAS Collaboration, 2012; The CMS Collaboration, 2012) and subsequent measurement of its properties. The bulk of the likelihood ratio tests at the LHC are based on the distribution of a single event-level feature that discriminates between a hypothesized process of interest (labeled *signal*) and various other processes (labeled *background*). Typically, data generated from the simulator are used to approximate the density at various parameter points, and an interpolation algorithm is used to approximate the parameterized model (Cranmer et al., 2012). In order to improve the statistical power of these tests, hundreds of these searches have already been using supervised learning to train classifiers to discriminate between two two discrete hypotheses based on a high dimensional feature vector $\mathbf{x}$. The results of this paper outline how to extend the use of discriminative classifiers for composite hypotheses (parametrized by $\theta$) in a way that fits naturally into the established likelihood based inference techniques.

The rest of the paper is organized as follows. In Section 2, we first introduce the likelihood ratio test statistic in the setting of simple hypothesis testing, and then outline how it can be computed exactly using calibrated classifiers. In Section 3, we generalize the proposed approach to the case of composite hypothesis testing and discuss directions for approximating the statistic efficiently. We then illustrate the proposed method in Section 4 and outline how it could improve statistical analysis within the field of high energy physics. Related work and conclusions are finally presented in Sections 5 and 6.

# 2 Likelihood ratio tests

## 2.1 Simple hypothesis testing

Let $\mathbf{X}$ be a random vector with values $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^p$ and let $p_{\mathbf{X}}(\mathbf{x}|\theta)$ denote the density probability of $\mathbf{X}$ at value $\mathbf{x}$ under the parameterization $\theta$. Let also assume i.i.d. observed data $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$. 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

$$\lambda(\mathcal{D}; \theta_0, \theta_1) = \prod_{\mathbf{x} \in \mathcal{D}} \frac{p_{\mathbf{X}}(\mathbf{x}|\theta_0)}{p_{\mathbf{X}}(\mathbf{x}|\theta_1)} \tag{2.1}$$

is the most powerful test statistic.

In order to evaluate $\lambda(\mathcal{D})$, one must be able to evaluate the probability densities $p_{\mathbf{X}}(\mathbf{x}|\theta_0)$ and $p_{\mathbf{X}}(\mathbf{x}|\theta_1)$ at any value $\mathbf{x}$. However, it is increasingly common in science that one has a complex simulation that can act as generative model for $p_{\mathbf{X}}(\mathbf{x}|\theta)$, but one cannot evaluate the density directly. For instance, this is the case in high energy physics (Neal, 2007) where the simulation of particle detectors can only be done in the forward mode.

## 2.2 Approximating likelihood ratios with classifiers

The main result of this paper is to generalize the observation that one can form a test statistic

$$\lambda'(\mathcal{D}; \theta_0, \theta_1) = \prod_{\mathbf{x} \in \mathcal{D}} \frac{p_{\mathbf{U}}(u = s(\mathbf{x})|\theta_0)}{p_{\mathbf{U}}(u = s(\mathbf{x})|\theta_1)} \tag{2.2}$$

that is strictly equivalent to 2.1, provided the change of variable $\mathbf{U} = s(\mathbf{X})$ is based on a (parameterized) function $s$ that is strictly monotonic with the density ratio

$$r(\mathbf{x}; \theta_0, \theta_1) = \frac{p_{\mathbf{X}}(\mathbf{x}|\theta_0)}{p_{\mathbf{X}}(\mathbf{x}|\theta_1)}. \tag{2.3}$$

As derived below, this allows to recast the original likelihood ratio test into an alternate form in which supervised learning can be used to build $s(\mathbf{x})$ as a discriminative classifier. In Section 3 we extend this result to generalized likelihood ratio tests, where it will be useful to have the classifier decision function $s$ parameterized in terms of $(\theta_0, \theta_1)$.

**Theorem 1.** *Let $\mathbf{X}$ be a random vector with values in $\mathcal{X} \subseteq \mathbb{R}^p$ and parameterized probability density $p_{\mathbf{X}}(\mathbf{x} = (x_1, ..., x_p)|\theta)$ and let $s : \mathbb{R}^p \mapsto \mathbb{R}$ be a function monotonic with the density ratio $r(\mathbf{x}; \theta_0, \theta_1)$, for given parameters $\theta_0$ and $\theta_1$. In these conditions,*

$$r(\mathbf{x}; \theta_0, \theta_1) = \frac{p_{\mathbf{X}}(\mathbf{x}|\theta_0)}{p_{\mathbf{X}}(\mathbf{x}|\theta_1)} = \frac{p_{\mathbf{U}}(u = s(\mathbf{x})|\theta_0)}{p_{\mathbf{U}}(u = s(\mathbf{x})|\theta_1)}, \tag{2.4}$$

*where $p_{\mathbf{U}}(u = s(\mathbf{x}; \theta_0, \theta_1)|\theta)$ is the induced probability density of $\mathbf{U} = s(\mathbf{X}; \theta_0, \theta_1)$.*

*Proof.* Starting from the definition of the probability density function, we have

$$\begin{aligned} p_{\mathbf{U}}(u = s(\mathbf{x})|\theta_0) &= \frac{d}{du} \int_{\{\mathbf{x}' \in \mathbb{R}^p : s(\mathbf{x}') \leq u\}} p_{\mathbf{X}}(\mathbf{x}'|\theta_0) d\mathbf{x}' \\ &= \int_{\mathbb{R}^p} \delta(u - s(\mathbf{x}')) p_{\mathbf{X}}(\mathbf{x}'|\theta_0) d\mathbf{x}' \end{aligned} \tag{2.5}$$

Intuitively, this expression can be understood as the integral over all $\mathbf{x}' \in \mathbb{R}^p$ such that $s(\mathbf{x}') = u$, as picked by the Dirac $\delta$ function. Given Theorem 6.1.5 of Hörmander (1990), it further comes

$$p_{\mathbf{U}}(u = s(\mathbf{x})|\theta_0) = \int_{\{\mathbf{x}' \in \mathbb{R}^p : s(\mathbf{x}') = u\}} \frac{1}{|\nabla s(\mathbf{x}')|} p_{\mathbf{X}}(\mathbf{x}'|\theta_0) dS_{\mathbf{x}'} \tag{2.6}$$

where $|\nabla s(\mathbf{x}')| = \sqrt{\sum_{i=1}^p |\frac{\partial}{\partial x_i} s(\mathbf{x}')|^2}$ and where $dS_{\mathbf{x}'}$ is the Euclidean surface measure on $\{\mathbf{x}' \in \mathbb{R}^p : s(\mathbf{x}') = u\}$. Also, since $s(\mathbf{x})$ is monotonic with $r(\mathbf{X}; \theta_0, \theta_1)$, there exists an invertible function $m : \mathbb{R}^+ \mapsto \mathbb{R}$ such that $s(\mathbf{x}) = m(r(\mathbf{X}; \theta_0, \theta_1))$. In particular, we have

$$\begin{aligned} \frac{p_{\mathbf{X}}(\mathbf{x}|\theta_0)}{p_{\mathbf{X}}(\mathbf{x}|\theta_1)} &= m^{-1}(s(\mathbf{x})) \\ p_{\mathbf{X}}(\mathbf{x}|\theta_0) &= m^{-1}(s(\mathbf{x})) p_{\mathbf{X}}(\mathbf{x}|\theta_1) \end{aligned} \tag{2.7}$$

Combining equations 2.6 and 2.7, the density ratio $r(\mathbf{X};\theta_0,\theta_1)$ can be pulled out of the integral, resulting in

$$\begin{aligned}
p_{\mathbf{U}}(u = s(\mathbf{x})|\theta_0) &= \int_{\{\mathbf{x}'\in\mathbb{R}^p : s(\mathbf{x}')=u\}} \frac{1}{|\nabla s(\mathbf{x}')|} m^{-1}(s(\mathbf{x}')) p_{\mathbf{X}}(\mathbf{x}'|\theta_1) dS_{\mathbf{x}'} \\
&= \int_{\{\mathbf{x}'\in\mathbb{R}^p : s(\mathbf{x}')=u\}} \frac{1}{|\nabla s(\mathbf{x}')|} m^{-1}(u) p_{\mathbf{X}}(\mathbf{x}'|\theta_1) dS_{\mathbf{x}'} \\
&= m^{-1}(s(\mathbf{x})) \int_{\{\mathbf{x}'\in\mathbb{R}^p : s(\mathbf{x}')=u\}} \frac{1}{|\nabla s(\mathbf{x}')|} p_{\mathbf{X}}(\mathbf{x}'|\theta_1) dS_{\mathbf{x}'} \\
&= \frac{p_{\mathbf{X}}(\mathbf{x}|\theta_0)}{p_{\mathbf{X}}(\mathbf{x}|\theta_1)} \int_{\{\mathbf{x}'\in\mathbb{R}^p : s(\mathbf{x}')=u\}} \frac{1}{|\nabla s(\mathbf{x}')|} p_{\mathbf{X}}(\mathbf{x}'|\theta_1) dS_{\mathbf{x}'}. \qquad (2.8)
\end{aligned}$$

Similarly, Equation 2.6 can be used to derive $p_{\mathbf{U}}(u = s(\mathbf{x})|\theta_1)$, finally yielding

$$\begin{aligned}
\frac{p_{\mathbf{U}}(u = s(\mathbf{x})|\theta_0)}{p_{\mathbf{U}}(u = s(\mathbf{x})|\theta_1)} &= \frac{p_{\mathbf{X}}(\mathbf{x}|\theta_0)}{p_{\mathbf{X}}(\mathbf{x}|\theta_1)} \frac{\int_{\{\mathbf{x}'\in\mathbb{R}^p : s(\mathbf{x}')=u\}} \frac{1}{|\nabla s(\mathbf{x}')|} p_{\mathbf{X}}(\mathbf{x}'|\theta_1) dS_{\mathbf{x}'}}{\int_{\{\mathbf{x}'\in\mathbb{R}^p : s(\mathbf{x}')=u\}} \frac{1}{|\nabla s(\mathbf{x}')|} p_{\mathbf{X}}(\mathbf{x}'|\theta_1) dS_{\mathbf{x}'}} \\
&= \frac{p_{\mathbf{X}}(\mathbf{x}|\theta_0)}{p_{\mathbf{X}}(\mathbf{x}|\theta_1)}. \qquad (2.9)
\end{aligned}$$

□

In light of this result, the likelihood ratio estimation problem can now be recast as a (probabilistic) classification problem, by noticing that the decision function

$$s^*(\mathbf{x}) = \frac{p_{\mathbf{X}}(\mathbf{x}|\theta_1)}{p_{\mathbf{X}}(\mathbf{x}|\theta_0) + p_{\mathbf{X}}(\mathbf{x}|\theta_1)}. \qquad (2.10)$$

modeled by a classifier trained to distinguish samples $\mathbf{x} \sim p_{\theta_0}$ from samples $\mathbf{x} \sim p_{\theta_1}$ satisfies the conditions of Theorem 1 (see Appendix A for further details). In other words, supervised learning yields a sufficient procedure for Theorem 1 to hold, guaranteeing that any *universally strongly consistent* algorithm can be used for learning $s^*$. Note however, that it is not a necessary procedure since Theorem 1 holds for any monotonic function $m$ of the density ratio, not only for $m(r(\mathbf{x})) = (1 + r(\mathbf{x}))^{-1}$. Equivalently, Theorem 1 shows that in the case that we learn a probabilistic classifier $s(\mathbf{x})$ which is imperfect up to a monotonic transformation of $r(\mathbf{x})$, then one can still resort to calibration (i.e., modeling $p_{\mathbf{U}}(u = s(\mathbf{x}))$) to compute $r(\mathbf{x})$ exactly. For this reason, the proposed method is expected to be more robust than directly using $(1 - s(\mathbf{x}))/s(\mathbf{x})$ as an approximate of $r(\mathbf{x})$ (which indeed converges towards $r(x)$ when $s(x)$ tends to $s^*(x)$).

## 2.3 Learning and calibrating $s$

In order for the proposed approach to be useful in the likelihood-free setting, we need to be able to approximate both $s(\mathbf{x})$ and $p(s(\mathbf{x})|\theta)$ based on a finite number of samples $\{\mathbf{x}_i\}$ drawn from the generative model $p(\mathbf{x}|\theta)$.

As outlined above, any consistent probabilistic classification algorithm can be used for learning an approximate map $\hat{s}(\mathbf{x})$ of Eqn. 2.10. In the common case where the density ratio is expected to smoothly vary around $\mathbf{x}$, we would however recommend learning models whose output value $\hat{s}(\mathbf{x})$ also smoothly varies around $\mathbf{x}$, such as neural networks. For small training sets, tree-based methods are not expected to work so well for this use case, since they usually model $\hat{s}(\mathbf{x})$ as a non-strictly monotonic composition of step functions. In such cases where $\hat{s}(\mathbf{x})$ is not monotonic with $r(\mathbf{x})$, the induced probability does not factorize as in Eqn. 2.8, leading to artifacts in the resulting approximation of the density ratio. Provided enough training data, accurate results can however still be achieved, given the universal approximator capacity of tree-based models.

Given a reduction map $s$, our results show that a statistic equivalent to the likelihood ratio can be constructed, provided $p(s(\mathbf{x})|\theta)$ can be evaluated. Again, we do not have a direct and exact way for evaluating this density, but an approximation $\hat{p}(\hat{s}(\mathbf{x})|\theta)$ can be built instead, e.g. using density estimation algorithms, such as histograms or kernel density estimation applied to $\{\hat{s}(\mathbf{x}_i)\}$, where the $\{\mathbf{x}_i\}$ are drawn from the generative model.

Most notably, learning such an approximation of $p(s(\mathbf{x})|\theta)$ is a much simpler problem than learning $p(\mathbf{x}|\theta)$, since the reduction $s$ projects $\mathbf{x}$ into a one-dimensional space in which only the (simpler) informative content of $r(\mathbf{x})$ is preserved.

An alternative approach for calibration is to approximate the density ratio $r(\hat{s}(\mathbf{x}))$ directly. For instance, isotonic regression, which is commonly used to transform the classifier score $\hat{s}(\mathbf{x})$ into $\hat{s}_{\text{iso}}(\mathbf{x})$ that more accurately reflect the posterior probability $s^*(\mathbf{x})$ of Eq. 2.10, can be used for calibration. This is done by inverting the relationship $r(\mathbf{x}) = (1 - s^*(\mathbf{x}))/s^*(\mathbf{x})$ to obtain $\hat{r}(\mathbf{x}) = (1 - \hat{s}_{\text{iso}}(\mathbf{x}))/\hat{s}_{\text{iso}}(\mathbf{x})$. **Gilles, should we add more about direct density ratio estimation (currently in 5. Related Work) here?**

One strength of the proposed approach is that it factorizes the approximation of the dimensionality reduction ($\hat{s}(\mathbf{x}) \approx s(\mathbf{x})$) from the calibration procedure ($\hat{p}(\hat{s}(\mathbf{x})|\theta) \approx p(\hat{s}(\mathbf{x})|\theta)$ or $\hat{r}(\hat{s}(\mathbf{x})) \approx r(\hat{s}(\mathbf{x}))$). Thus, even if the classifier does a poor job at learning the optimal decision function 2.10 and therefore at reproducing the level sets of the per-sample likelihood ratio, the density of $\hat{s}$ can still be well calibrated. In that case, one might loose power, but the resulting inference will still be valid. This point was made by Neal (2007) and is well appreciated by the particle physics community that typically takes a conservative attitude towards the use of machine learning classifiers precisely due to concerns about the calibration of $p$-values in the face of nuisance parameters associated to the simulator.

# 3 Generalized likelihood ratio tests

Thus far we have shown that the target likelihood ratio $r(\mathbf{x}; \theta_0, \theta_1)$ with high dimensional features $\mathbf{x}$ can be reproduced via the univariate densities $p(s(\mathbf{x})|\theta_0)$ and $p(s(\mathbf{x})|\theta_1)$ if the reduction $s(\mathbf{x})$ is monotonic with $r(\mathbf{x}; \theta_0, \theta_1)$. We now generalize from the ratio of two simple hypotheses specified by $\theta_0$ and $\theta_1$ to the case of composite hypothesis testing where

$\theta$ are continuous model parameters.

## 3.1 Composite hypothesis testing

In the case of composite hypotheses $\theta \in \Theta_0$ against an alternative $\theta \in \Theta_1$ (such that $\Theta_0 \cap \Theta_1 = \emptyset$ and $\Theta_0 \cup \Theta_1 = \Theta$), the generalized likelihood ratio test, also known as the profile likelihood ratio test, is commonly used

$$\Lambda(\Theta_0) = \frac{\sup_{\theta \in \Theta_0} p(\mathcal{D}|\theta)}{\sup_{\theta \in \Theta} p(\mathcal{D}|\theta)} \, . \tag{3.1}$$

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.

Evaluating the generalized likelihood ratio as defined by Eqn. 3.1 requires finding for both the numerator and the denominator the maximum likelihood estimator

$$\hat{\theta} = \arg\max_{\theta} p(\mathcal{D}|\theta). \tag{3.2}$$

Again, this is made difficult in the likelihood-free setting and it is not obvious that we can find the same estimators if we are working instead with $p(s(\mathbf{x})|\theta)$. Fortunately, there is a construction based on $s$ that works: the maximum likelihood estimate of Eqn. 3.2 is the same as the value that maximizes the likelihood ratio with respect to $p(\mathcal{D}|\theta_1)$, for some fixed value of $\theta_1$ chosen such that the support of $p(\mathbf{x}|\theta_1)$ covers the support of $p(\mathbf{x}|\theta)$. This allows us to use Theorem 1 to reformulate the maximum likelihood estimate as

$$\begin{aligned}\hat{\theta} &= \arg\max_{\theta} p(\mathcal{D}|\theta) \\ &= \arg\max_{\theta} \prod_{\mathbf{x}\in\mathcal{D}} \frac{p(\mathbf{x}|\theta)}{p(\mathbf{x}|\theta_1)} \\ &= \arg\max_{\theta} \prod_{\mathbf{x}\in\mathcal{D}} \frac{p(s(\mathbf{x};\theta,\theta_1)|\theta)}{p(s(\mathbf{x};\theta,\theta_1)|\theta_1)} \ , \end{aligned} \tag{3.3}$$

where $s(\mathbf{x};\theta,\theta_1)$ denotes a *parameterized* transformation $s$ of $\mathbf{X}$ in terms of $(\theta,\theta_1)$ that is monotonic with $r(\mathbf{x};\theta,\theta_1)$. Note that it is important that we include the denominator $p(s(\mathbf{x};\theta,\theta_1)|\theta_1)$ because this cancels Jacobian factors that vary with $\theta$.

Finally, once the maximum likelihood estimates have been found for both the numerator and denominator of Eqn. 3.1, the generalized likelihood ratio can be estimated as outlined in Section 2.2 for simple hypothesis testing.

## 3.2 Parameterized classification

In order to provide parameter inference in the likelihood-free setting as described above, we must train a family $s(\mathbf{x};\theta_0,\theta_1)$ of classifiers parameterized 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 $\theta_1$, using the procedure outlined in Section 2, it is desirable and convenient to have a smooth evolution of the classification score as a function of the parameters. For this reason, we anticipate a single learning stage based on training data with input $(\mathbf{x},\theta_0,\theta_1)_i$ and target $y_i$, as outlined in Algorithm 1. Somewhat unusually, the unknown values of the parameters are taken as input to the classifier; their values will be specified via the enveloping (generalized) likelihood ratio of Eqn. 3.1. In this way, the parameterized classifier now models the distribution of the output $y$ conditional to $(\mathbf{x},\theta_0,\theta_1)$, for any $\mathbf{x}$ and any combination of parameter values $\theta_0,\theta_1$.

**Algorithm 1** Learning a parameterized classifier.

```
$\mathcal{T} := \{\}$;
while size($\mathcal{T}$) $< N$ do
    Draw $\theta_0 \sim \pi_{\Theta_0}$;
    Draw $\mathbf{x} \sim p(\mathbf{x}|\theta_0)$;
    $\mathcal{T} := \mathcal{T} \cup \{((\mathbf{x}, \theta_0, \theta_1), y = 0)\}$;
    Draw $\theta_1 \sim \pi_{\Theta_1}$;
    Draw $\mathbf{x} \sim p(\mathbf{x}|\theta_1)$;
    $\mathcal{T} := \mathcal{T} \cup \{((\mathbf{x}, \theta_0, \theta_1), y = 1)\}$;
end while
Learn a single classifier $s(\mathbf{x}; \theta_0, \theta_1)$ from $\mathcal{T}$.
```

While the optimal decision function 2.10 is expected to be learned for the parameter values $\theta_0$ and $\theta_1$ selected in Algorithm 1, it is not clear whether the optimal decision function can be expected for data generated from $\theta'_0$ and $\theta'_1$ never jointly encountered during learning. Similarly, it is not clear how the limited capacity of the classifier may impact the performance the resulting parameterized decision function. Preliminary exploration by Baldi et al. (2016) shows that a uniform grid scan over parameter space is an effective practical approach; however, we introduce the distributions $\pi_{\Theta_0}$ and $\pi_{\Theta_1}$ into the Algorithm 1 to allow for a more sophisticated sampling strategy.

## 3.3 Parameterized calibration

Once the parametrized classifier $\hat{s}(\mathbf{x}; \theta_0, \theta_1)$ is trained, we can use the generative model together with one of the calibration strategies discussed in Sec. 2.3 for particular values of $\theta_0$ and $\theta_1$. For a single parameter point $\theta$, this is a tractable univariate density estimation problem. The challenge comes from the need to calibrate this density for all values of $\theta$. A straightforward approach would be to run the generative model on demand for any particular value of $\theta$. In the context of a likelihood fit this would mean that the optimization algorithm that is trying to maximize the likelihood with respect to $\theta$ needs access to the generative model $p(\mathbf{x}|\theta)$. This is the strategy used for the examples presented in Sec. 4.

Calibrating the density on-demand can be impractical when the generative model is computationally expensive or has high-latency (for instance some human intervention is required to reconfigure the generative model). In high energy physics, where it is common to calibrate the distribution of a fixed classifier. There the strategy is to interpolate the distribution between discrete values of $\theta$ in order to produce a continuous parameterization for $p(s|\theta)$ (Read, 1999; Cranmer et al., 2012; Baak et al., 2015). One can easily imagine a number of approaches to parametrized calibration and the relative merits of those approaches will depend critically on the dimensionality of $\theta$ and the computational cost of the generative model. We leave a more general strategy for this overarching optimization problem as an area of future work.

## 3.4 Mixture models

In the special case of (simple or composite) hypothesis testing between models defined as known mixtures of several components, i.e. when $p(\mathbf{x}|\theta)$ can be written as

$$p(\mathbf{x}|\theta) = \sum_c w_c(\theta) p_c(\mathbf{x}|\theta), \tag{3.4}$$

the target likelihood ratio can be formulated in terms of pairwise classification problems. Specifically, we can write

$$\begin{aligned}\frac{p(\mathbf{x}|\theta_0)}{p(\mathbf{x}|\theta_1)} &= \frac{\sum_c w_c(\theta_0) p_c(\mathbf{x}|\theta_0)}{\sum_{c'} w_{c'}(\theta_1) p_{c'}(\mathbf{x}|\theta_1)} \\ &= \sum_c \left[ \sum_{c'} \frac{w_{c'}(\theta_1)}{w_c(\theta_0)} \frac{p_{c'}(\mathbf{x}|\theta_1)}{p_c(\mathbf{x}|\theta_0)} \right]^{-1} \\ &= \sum_c \left[ \sum_{c'} \frac{w_{c'}(\theta_1)}{w_c(\theta_0)} \frac{p_{c'}(s_{c,c'}(\mathbf{x};\theta_0,\theta_1)|\theta_1)}{p_c(s_{c,c'}(\mathbf{x};\theta_0,\theta_1)|\theta_0)} \right]^{-1} . \end{aligned} \tag{3.5}$$

The second line is a trivial, but a useful decomposition into pairwise density ratio sub-problems between $p_{c'}(\mathbf{x}|\theta_1)$ and $p_c(\mathbf{x}|\theta_0)$. The third line uses Theorem 1 to relate the high-dimensional likelihood ratio into an equivalent calibrated likelihood ratio based on the univariate density of the corresponding classifier.

In applications where mixture models are commonly used, this decomposition allows one to construct better likelihood ratio estimates since it allows the classifiers $s_{c,c'}$ to focus on simpler sub-problems, for which higher accuracy is expected.

Finally, as a technical point, in the situation where the only free parameters of the model are the mixture coefficients $w_c$, the distributions $p_c(s_{c,c'}(\mathbf{x};\theta_0,\theta_1)|\theta)$ are independent of $\theta$. For this reason, sub-ratios $r_{c,c'}(\mathbf{x};\theta_0,\theta_1) = \frac{p_{c'}(s_{c,c'}(\mathbf{x};\theta_0,\theta_1)|\theta_1)}{p_c(s_{c,c'}(\mathbf{x};\theta_0,\theta_1)|\theta_0)}$ simplify to $\frac{p_{c'}(s_{c,c'}(\mathbf{x}))}{p_c(s_{c,c'}(\mathbf{x}))}$, which can be pre-computed without the need of parameterized classification or calibration.

# 4 Examples and applications

In this section, we illustrate the proposed method on two representative examples where the exact likelihood is known and then discuss its application to high energy physics. The code used to produce the results and extended details for these examples is available in Ref. (Cranmer et al., 2016).

## 4.1 Likelihood ratios of mixtures of normals

As a simple and illustrative example, let us first consider the approximation of the log-likelihood ratio $\log\left(r(\mathbf{x}; \gamma = 0.05, \gamma = 0)\right)$ between the 1D mixtures $p(\mathbf{x}|\gamma = 0.05)$ and $p(\mathbf{x}|\gamma = 0)$ defined as

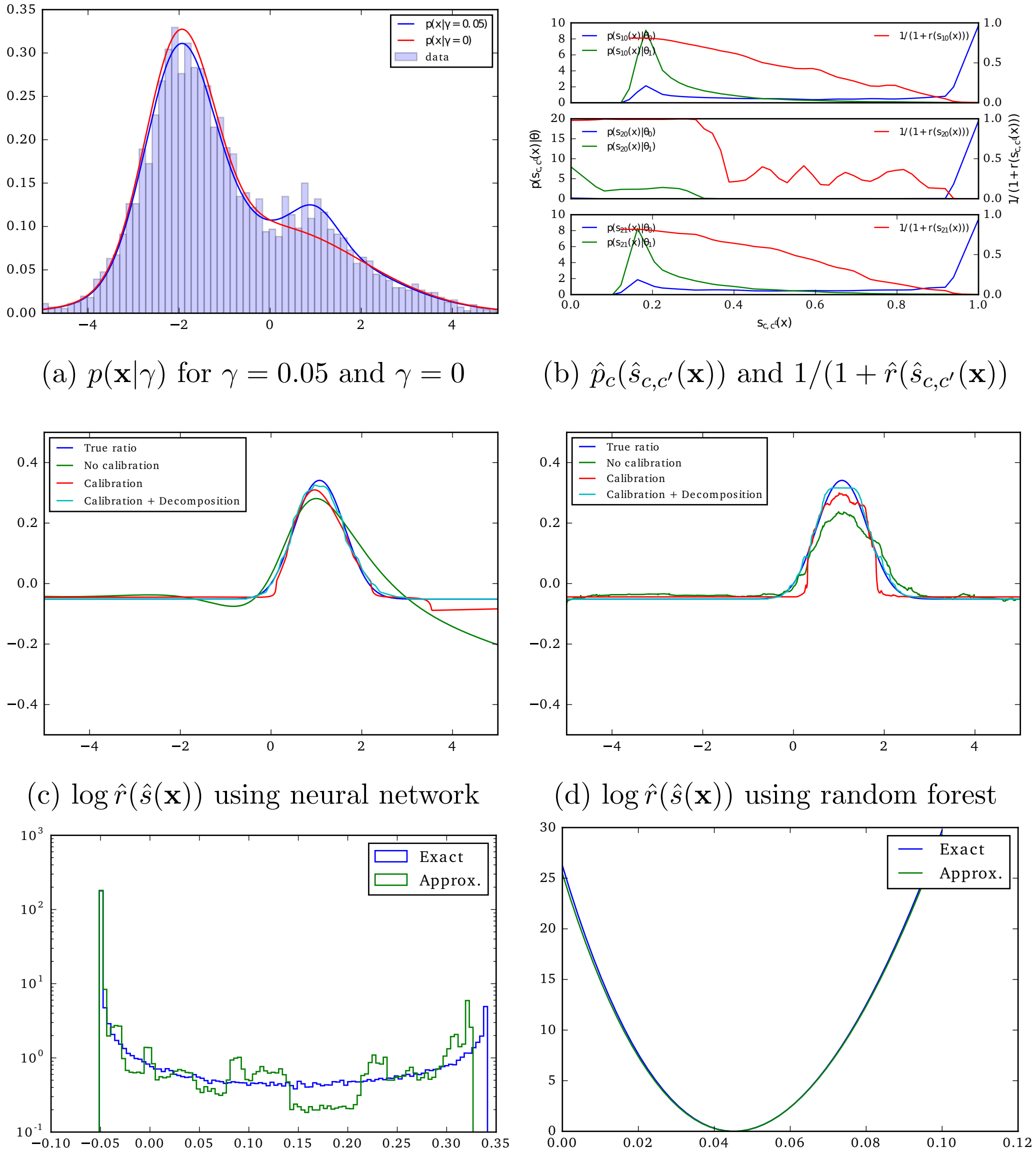
$$p(\mathbf{x}|\gamma) = (1-\gamma)\frac{p_{c_0}(\mathbf{x}) + p_{c_1}(\mathbf{x})}{2} + \gamma\, p_{c_2}(\mathbf{x}), \tag{4.1}$$

where $p_{c_0} := \mathcal{N}(\mu = -2, \sigma^2 = 0.25^2)$, $p_{c_1} := \mathcal{N}(\mu = 0, \sigma^2 = 4)$, $p_{c_2} := \mathcal{N}(\mu = 1, \sigma^2 = 0.25)$. Samples drawn for the nominal value $\gamma = 0.05$ are shown in Fig. 1a and used later for inference.

Fig. 1b shows the intermediate stages for the decomposition described in Section 3.4. The blue and green curves show $p_{c'}(\hat{s}_{c,c'}(\mathbf{x}))$, the distributions for the score for the sub-classifiers for the three pair-wise comparisons of the mixture components. The red curves in Fig. 1b show the approximation of the density ratio obtained from those distributions.

Figures 1c and 1d show the approximate $\log\left(\hat{r}(\mathbf{x})\right)$ as a function of $\mathbf{x}$ using a 2-layer neural network and a random forest for the classifier $\hat{s}(\mathbf{x})$. The neural network provides a smoother approximation, while the random has some artifacts due to the fact that the decision function is piece-wise constant. The blue curves show the exact log-ratio, the green curves show $\log\left((1 - \hat{s}(\mathbf{x}))/\hat{s}(\mathbf{x})\right)$ without calibration, while the red curve is the improved approximation $\log \hat{r}(\mathbf{x})$ calibrated using histograms. Finally, the cyan curve shows the approximated log-likelihood ratio when decomposing the mixture, as seen in Fig. 1b. By leveraging the fact that densities are mixtures, the capacity of the underlying classifiers can be more effectively focused on easier classification tasks, resulting as expected in even more accurate approximations.

As the results show, calibrating $\hat{s}(\mathbf{x})$ through univariate density estimation of $\hat{p}(\hat{s}(\mathbf{x}))$ is key to obtaining accurate results. Standard histograms with uniform binning have been



(a) $p(\mathbf{x}|\gamma)$ for $\gamma = 0.05$ and $\gamma = 0$

(b) $\hat{p}_c(\hat{s}_{c,c'}(\mathbf{x}))$ and $1/(1 + \hat{r}(\hat{s}_{c,c'}(\mathbf{x}))$

(c) $\log \hat{r}(\hat{s}(\mathbf{x}))$ using neural network

(d) $\log \hat{r}(\hat{s}(\mathbf{x}))$ using random forest

(e) $p(\log \hat{r}(\hat{s}(\mathbf{x})) \,|\, \gamma = 0.05)$

(f) $-\log \Lambda(\gamma)$

Figure 1: Approximation of the log-likelihood ratio $\log r(\mathbf{x}; \gamma = 0.05, \gamma = 0)$ with calibrated classifiers.

used here for illustrative purposes, but we anticipate that more sophisticated calibration strategies will be important in further development of this method. We leave this as an area for future study.

Fig. 1e shows the distribution of $\log \hat{r}(\mathbf{x}))$ for $\gamma = 0.05$ using the decomposed approximation of $\hat{r}(\mathbf{x}))$ with neural networks and the distribution of the exact log-likelihood ratio. While there are some artifacts in the distribution in the low-probability regions and the maximum value of the log-likehood ratio is underestimated, the overall shape of the distribution is well approximated.

Finally, we come to the log-likelihood curve

$$\log \Lambda(\gamma) = \log \frac{p(\mathcal{D}|\gamma)}{\sup_{\gamma \in \Theta} p(\mathcal{D}|\gamma)} \tag{4.2}$$

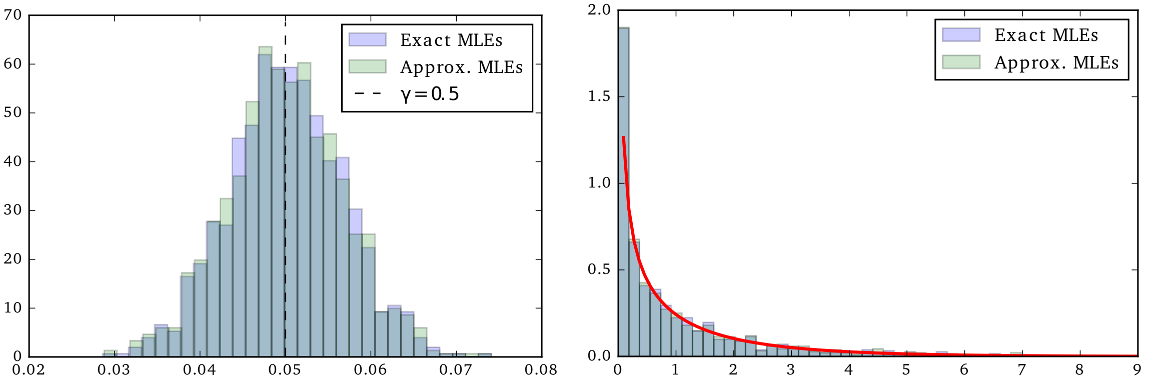
for the dataset $\mathcal{D}$ shown in Fig. 1a. By exploiting the fact that

$$\log \frac{p(\mathcal{D}|\gamma)}{\sup_{\gamma \in \Theta} p(\mathcal{D}|\gamma)} = \log \frac{p(\mathcal{D}|\gamma)}{p(\mathcal{D}|\gamma = 0)} - \log \frac{\sup_{\gamma \in \Theta} p(\mathcal{D}|\gamma)}{p(\mathcal{D}|\gamma = 0)}, \tag{4.3}$$

the generalized likelihood ratio can be computed by evaluating both terms with respect to a common reference $\gamma = 0$ as outlined in Section 3. Fig. 1f shows that the exact likelihood curve is very well approximated by the method, confirming that even when the raw classifier does a poor job at modeling the $s^*(\mathbf{x})$, a good approximations of the likelihood ratio can still be obtained by calibrating $s(\mathbf{x})$ (and by decomposing the mixture, if possible).

An advantage of this approach compared to ABC is that the classifier and calibration – computationally intensive parts of the approximation – are independent of the of the dataset $\mathcal{D}$. Thus once trained and calibrated, the approximation can be applied to any dataset $\mathcal{D}$. This makes it computationally efficient to perform ensemble tests of the method.

Fig. 2a shows the empirical distribution of the maximum likelihood estimators (MLEs) from the approximate likelihood compared to the distribution of the MLEs from the exact



(a) True MLEs vs. approximated MLEs.

(b) $p(-2\log\Lambda(\gamma = 0.05)\,|\,\gamma = 0.05)$

Figure 2: Using approximated likelihood ratios for parameter inference yields an unbiased maximum likelihood estimator $\hat{\gamma}$, as empirically estimated 1000 times over re-sampled observed data.

likelihood. It clearly demonstrates that in this case the approximate likelihood yields an unbiased estimator with the essentially the same variance as the exact MLE. In addition to the MLE, we can study the coverage of a confidence interval based on the likelihood ratio test statistic. This is done by evaluating $-2\log\Lambda(\gamma = 0.05)$ for samples drawn from $p(\mathbf{x}|\gamma = 0.05)$. Wilks's theorem states that the distribution of $-2\log\Lambda(\gamma = 0.05)$ should follow a $\chi^2_1$ distribution. Fig. 2b also confirms this behavior, supporting the applicability of this method for likelihood-based inference techniques in the likelihood-free setting.

## 4.2 Parameterized inference from multidimensional data

Let us now consider the more challenging problem of likelihood-free inference with multidimensional data. For the sake of the illustration, we will assume 5-dimensional feature $\mathbf{x}$ generated from the following process $p_0$:

1. $\mathbf{z} := (z_0, z_1, z_2, z_3, z_4)$, such that $z_0 \sim \mathcal{N}(\mu = \alpha, \sigma = 1)$, $z_1 \sim \mathcal{N}(\mu = \beta, \sigma = 3)$, $z_2 \sim \text{Mixture}(1/2\,\mathcal{N}(\mu = -2, \sigma = 1), 1/2\,\mathcal{N}(\mu = 2, \sigma = 0.5))$, $z_3 \sim \text{Exponential}(\lambda = 3)$, and $z_4 \sim \text{Exponential}(\lambda = 0.5)$;
2. $\mathbf{x} := R\mathbf{z}$, where $R$ is a fixed semi-positive definite $5 \times 5$ matrix defining a fixed projection of $\mathbf{z}$ into the observed space.

The observations $\mathcal{D}$ represented in Fig. 3 are random samples with $\alpha = 1$ and $\beta = -1$. Our goal is to infer the values $\alpha$ and $\beta$ based on $\mathcal{D}$. We construct the log-likelihood ratio

$$-2 \log \Lambda(\alpha, \beta) = -2 \log \frac{p(\mathcal{D}|\alpha, \beta)}{\sup_{\alpha,\beta} p(\mathcal{D}|\alpha, \beta)} \tag{4.4}$$

that we calculate by exploiting the fact that

$$\log \frac{p(\mathcal{D}|\alpha, \beta)}{\sup_{\alpha,\beta} p(\mathcal{D}|\alpha, \beta)} = \log \frac{p(\mathcal{D}|\alpha, \beta)}{p(\mathcal{D}|\alpha = 0, \beta = 0)} - \log \frac{\sup_{\alpha,\beta} p(\mathcal{D}|\alpha, \beta)}{p(\mathcal{D}|\alpha = 0, \beta = 0)}. \tag{4.5}$$

Following the procedure described in Section 3.2, we build a build single 2-layer neural network (with 5+2 inputs and one output node) to form the parameterized classifier $s(\mathbf{x}; \theta_0, \theta_1)$ and fix $\theta_1 = (\alpha = 0, \beta = 0)$. Since the generative model is not expensive, the classifier output is calibrated on-the-fly with histograms for every candidate parameter pair $(\alpha, \beta)$.

Figure 4a shows the exact log-likelihood ratio for this dataset, which has an exact MLE at $(\hat{\alpha} = 1.012, \hat{\beta} = -0.9221)$. Figure 4b shows the approximate log-likelihood ratio evaluated on a coarse grid of parameter values. Some roughness in the contours is observed, which is due to primarily to variance introduced in the calibration procedure. In addition to the statistical fluctuations due to finite calibration samples, there are also fluctuations introduced from changes in the binning of the calibration histograms as $\alpha$ and $\beta$ vary. As discussed in Section 3.3, a parameterized calibration procedure should ameliorate this issue, but that is left for now as an area for future work. Nevertheless, optimizing the approximate

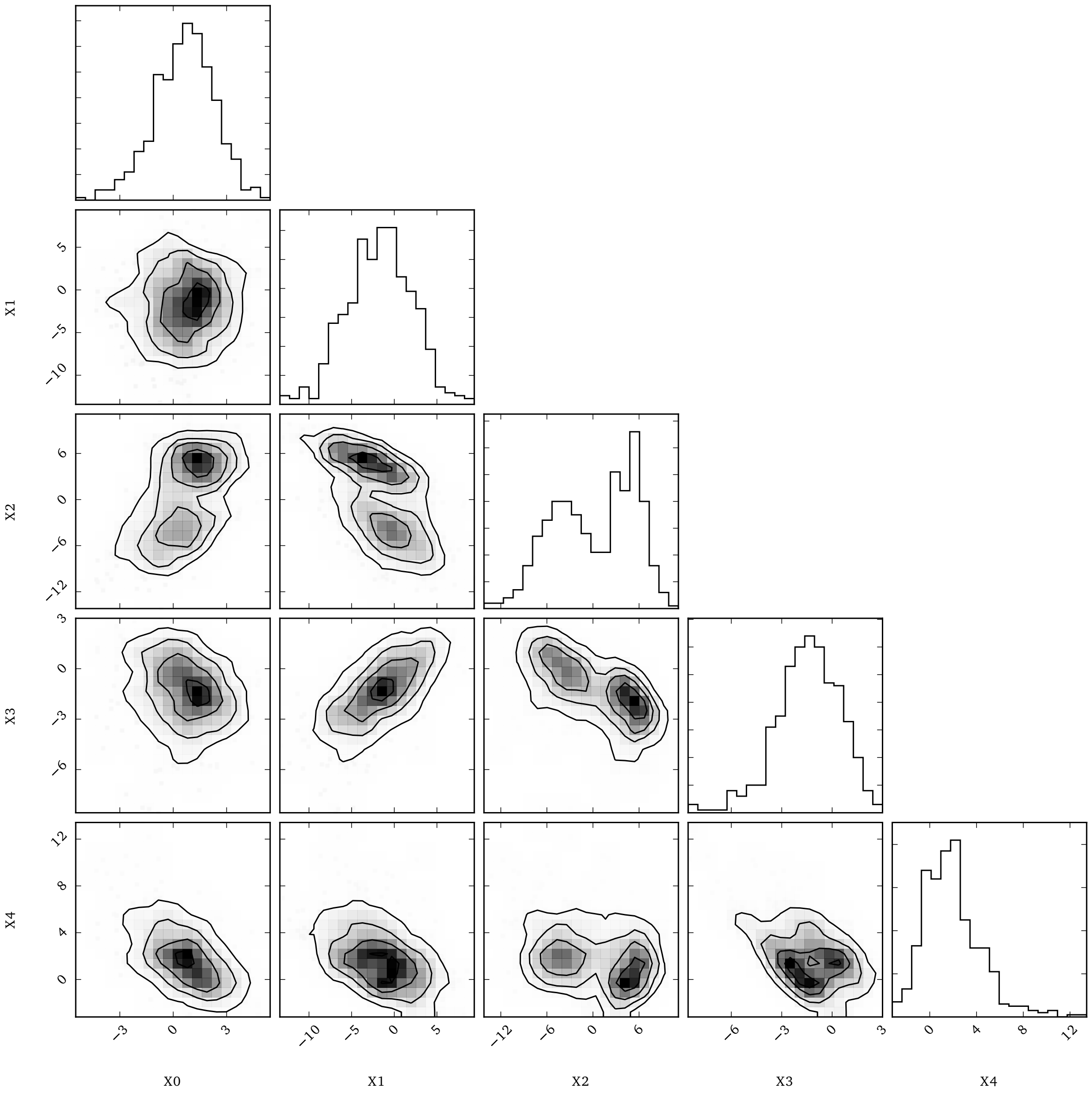
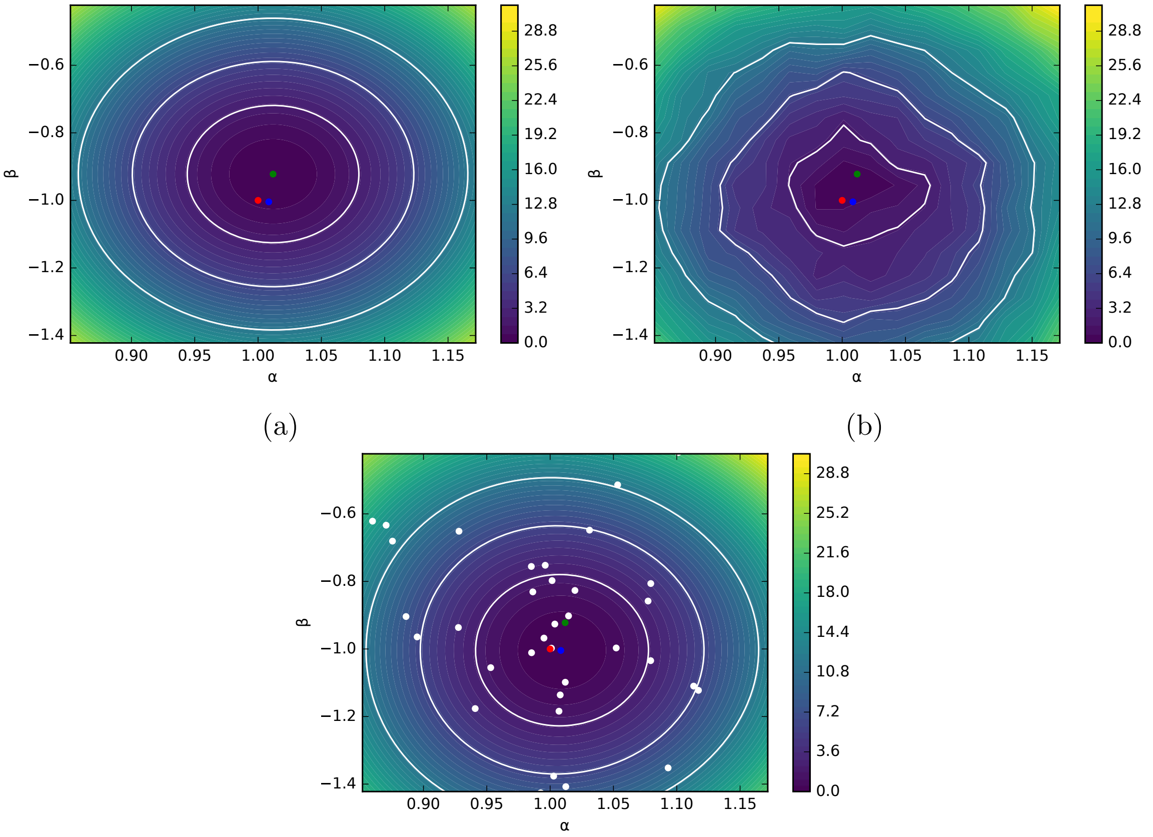


Figure 3: Observed 5-dimensional data $\mathcal{D}$ for nominal values $(\alpha = 1, \beta = -1)$. The size of $\mathcal{D}$ is set to 500.



(a)

(b)

(c)

Figure 4: Inference from exact and approximate likelihood ratios. The red dot corresponds to the true values ($\alpha = 1, \beta = -1$) used to generate $\mathcal{D}$, the green dot is the MLE ($\hat{\alpha} = 1.012, \hat{\beta} = -0.9221$) from the exact likelihood, while the blue dot is the MLE ($\hat{\alpha} = 1.008, \hat{\beta} = -1.004$) from the approximate likelihood. 1, 2 and 3-$\sigma$ contours are shown in white. (4a) The exact log-likelihood for the observed data $\mathcal{D}$. (4b) The approximate log-likelihood evaluated on a coarse $15 \times 15$ fixed grid. (4c) A Gaussian Process surrogate of the approximate log-likelihood ratio estimated from a Bayesian optimization procedure. White dots show the parameter points sampled during the optimization process.

log-likelihood ratio with a Bayesian optimization (Brochu et al., 2010) procedure is efficient and effective. After 50 likelihood evaluations, the maximum likelihood estimate is found at ($\hat{\alpha} = 1.008, \hat{\beta} = -1.004$). While the goal of the Bayesian optimization procedure is to find the maximum likelihood, the posterior mean of the internal Gaussian process, shown in Fig. 4c, is close to the exact log-likelihood ratio illustrated in Fig. 4a. In each case, the true values $\alpha = 1$ and $\beta = -1$ are contained within the $1 - \sigma$ likelihood contour.

Overall, this example further illustrates and confirms the ability of the proposed method for inference with multiple parameters and multi-dimensional data where reliable approximations $\hat{p}(\mathbf{x}|\theta_0)$ and $\hat{p}(\mathbf{x}|\theta_1)$ are often difficult to construct.

## 4.3 High energy physics

High energy physics was the original scientific domain that motivated the development of this procedure. In high energy physics, we are often searching for 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 $\mathbf{x}$, with corresponding distributions $p_s(\mathbf{x}|\nu)$ for signal and $p_b(\mathbf{x}|\nu)$ for background, where $\nu$ are nuisance parameters describing uncertainties in the underlying physics prediction or response of the measurement device. The total model is a mixture of the signal and background, and $\mu$ is the mixture coefficient associated to the signal component, that is

$$p(\mathcal{D}|\mu, \nu) = \prod_{\mathbf{x} \in \mathcal{D}} \left[ \mu p_s(\mathbf{x}|\nu) + (1 - \mu) p_b(\mathbf{x}|\nu) \right] \ . \tag{4.6}$$

Accordingly, new particle searches at the LHC are typically framed as hypothesis tests where the null corresponds to $\mu = 0$, and the generalized likelihood ratio is used as a test statistic.

Nuisance parameters are an after thought in the typical usage of machine learning in high energy physics. The classifiers are typically trained with data generated using a fixed nominal value of the nuisance parameters $\nu = \nu_0$. However, as experimentalists we know that we must account for the systematic uncertainties that correspond to the nuisance parameters $\nu$. Thus, typically we take the classifier $\hat{s}(\mathbf{x})$ as fixed and then propagate uncertainty by estimating $\hat{p}_s(\hat{s}(\mathbf{x})|\nu)$ with a parametrized calibration procedure. However, this classifier is clearly not optimal for $\nu \neq \nu_0$. In contrast, a parameterized classifier proposed in this work would yield more accurate estimates of the generalized likelihood ratio.

In addition to robustness to systematic uncertainties incorporated by the nuisance parameters $\nu$, the proposed method can be used to infer parameters of interest. Not only can the mixture coefficient $\mu$ be inferred using the decomposition procedure, but also physical parameters like particle masses that change the distribution of $\mathbf{x}$. This formalism represents a significant step forward in the usage of machine learning in high energy physics, where classifiers have always been used between two static classes of events and not parameterized explicitly in terms of the physical quantities we wish to measure.

Another approach for parameter inference with multi-dimensional data specific to high energy physics is the so-called matrix element method, in which one directly computes an approximate likelihood ratio by performing a computationally intensive integral associated to a simplified detector response (Volobouev, 2011). In the approach considered in this paper, the detailed detector response is naturally incorporated by the simulator; however, that integral is intractable for the matrix element method. Even with drastic simplifications of the detector response, the matrix element method can take several minutes of CPU time to calculate the likelihood ratio for a single event $\mathbf{x}$. The work here can be seen as aiming at the same conceptual target, but relying on machine learning to overcome the complexity of the detector simulation. It also offers enormous speed increase for evaluating the likelihood at the cost of an initial training stage. In practice, the matrix element method has only been used for searches and measurement of a single physical parameter (sometimes with a single nuisance parameter as in (Aaltonen et al., 2010)).

Contemporary examples where the technique presented here could have major impact include the measurement of coefficients to quantum mechanical operators describing the production and decay of the Higgs boson (Chen et al., 2015) and, if we are so lucky, measurement of the mass of supersymmetric particles in cascade decays (Allanach et al., 2000). Both of these examples involve data sets with many events, each with a feature vector $\mathbf{x}$ that has on the order of 10 components, and a parameter vector $\theta$ with 2-10 parameters of interest and possibly many more nuisance parameters.

# 5 Related work

The closest work to the proposed method is due to Neal (2007), who similarly considers the problem of approximating the likelihood function when only a generative model is available. That work sketches a scheme in which one uses a classifier with both $\mathbf{x}$ and $\theta$ as an input to serve as a dimensionality reduction map. The key distinction comes in the handling of $\theta$. Neal argues that a classifier cannot be used on real data, since we do not know the correct value for $\theta$, and goes on to outline an approach where one uses regression on a per-event basis to estimate $\hat{\theta}(\mathbf{x})$ and perform the composition $s(\mathbf{x}; \hat{\theta}(\mathbf{x}))$. As pointed out by the author, this can lead to a significant loss of information since a single observation $\mathbf{x}$ may carry little information about the true value of $\theta$, though a full data set $\mathcal{D}$ may be informative. The work of Neal (2007) correctly identifies this as an approximation of the target likelihood even in the case of a ideal classifier. In contrast, the approach described here does not eliminate the dependence of the classifier on $\theta$. Instead, we embed a parameterized classifier into the likelihood and postpone the evaluation of the classifier to the point of evaluation of the likelihood when $\theta$ is explicitly being tested. This avoids the loss of information that occurs from the regression step $\hat{\theta}(\mathbf{x})$ proposed by Neal (2007) and leads to Theorem 1, which is an exact result in the case of an ideal classifier. In both cases, the quality of the classifier is factorized from the calibration of its density, which allows for valid inference even if there is a loss of power due to a non ideal classifier.

Also close to our work, Scott and Nowak (2005) and Xin Tong (2013) consider the machine learning problem associated to Neyman-Pearson hypothesis testing. In a similar setup, they consider the situation where one does not have access to the underlying distributions, but only has i.i.d. samples from each hypothesis. This work generalizes that goal from the Neyman-Pearson setting to generalized likelihood ratio tests and emphasizes the connection with classification. Ihler et al. (2004) take on a different problem (tests of statistical independence) by using machine learning algorithms to find scalar maps from the high-dimensional feature space that achieve the desired statistical goal when the fundamental high-dimensional test is intractable.

More generally, likelihood ratio testing directly relates to the density ratio estimation problem, which consists in estimating the ratio of two densities from finite collections of observations $\mathcal{D}_0$ and $\mathcal{D}_1$. Density ratio estimation is connected to many machine learning fundamental problems, including transfer learning (Sugiyama and Kawanabe, 2012), probabilistic classification and regression (Vapnik, 1998), outlier detection (Hido et al., 2011), and many others. For learning under covariate shift, Shimodaira (2000) and Sugiyama and Müller (2005) estimate the density ratio $r(\mathbf{x}; \theta_0, \theta_1)$ from straightforward approximations $\hat{p}(\mathbf{x}|\theta_0)$ and $\hat{p}(\mathbf{x}|\theta_1)$ separately obtained using kernel density estimation. Despite its theoretical consistency, this approach is known to be ineffective in practice (Sugiyama et al., 2007; Bickel et al., 2009), since it relies on modeling numerator and denominator high-dimensional densities, which is a harder problem than modeling their ratio only. While the proposed method also proceeds in two similar steps, estimating $p(s(\mathbf{x}))$ is much easier than estimating $p(\mathbf{x})$, since $s$ projects $\mathbf{x}$ into a one-dimensional space in which only the informative content of $r(\mathbf{x})$ is preserved. Finally, in contrast with the proposed method which decouples reduction from calibration, other approaches proposed within the literature (see Sugiyama et al. (2012); Gretton et al. (2009); Nguyen et al. (2010); Vapnik et al. (2013) and references therein) provide solutions for estimating $r(\mathbf{x}; \theta_0, \theta_1)$ directly from $\mathbf{x}$, in one step. Under some assumptions, the convergence of the obtained estimates is also proven for some of these approaches.

# 6 Conclusions

In this work, we have outlined an approach to reformulate generalized likelihood ratio testing over a high-dimensional data set in terms of a univariate density of a classifier score. We have shown that a parameterized family of discriminative classifiers $\hat{s}(\mathbf{x}; \theta_0, \theta_1)$ trained and calibrated with a simulator can be used to approximate the likelihood ratio, even when it is not possible to directly evaluate the likelihood $p(\mathbf{x}|\theta)$. The proposed method offers an alternative to approximate Bayesian computation for parameter inference in the likelihood-free setting that can also be used in the frequentist formalism without specifying a prior over the parameters. A strength of this approach is that it separates the quality of the approximation of the target likelihood from the quality of the calibration. The former is related to the ability of supervised learning approaches to classification, which will continue to improve. The calibration procedure for a particular parameter point is fairly straightforward since it involves estimating a univariate density using a generative model of the data. The difficulty of the calibration stage is performing this calibration continuously in $\theta$. Different strategies to this calibration are anticipated depending on the dimensionality of $\theta$, the complexity of the resulting likelihood function, or the practical issues associated to running the simulator.

## Acknowledgments

KC and GL are both supported through NSF ACI-1450310. KC would like to thank Daniel Whiteson for encouragement and Alex Ihler for challenging discussions that led to a reformulation of the initial idea. KC would also like to thank Shimon Whiteson, Babak Shahbaba for advice in the presentation, Radford Neal for discussion of his earlier work, Yann LeCun, Philip Stark, and Pierre Baldi for their feedback on the project early in its conception, Balázs Kégl for feedback to the draft, and Yuri Shirman for reassuring cross checks of the Theorem. KC is grateful to UC-Irvine for their hospitality while this research was carried out and the Moore and Sloan foundations for their generous support of the data science environment at NYU.

## References

Aaltonen, T. et al. (2010). Top Quark Mass Measurement in the Lepton + Jets Channel Using a Matrix Element Method and *in situ* Jet Energy Calibration. *Phys.Rev.Lett.*, 105:252001.

Agostinelli, S. et al. (2003). GEANT4: A Simulation toolkit. *Nucl.Instrum.Meth.*, A506:250–303.

Allanach, B., Lester, C., Parker, M. A., and Webber, B. (2000). Measuring sparticle masses in nonuniversal string inspired models at the LHC. *JHEP*, 0009:004.

Baak, M., Gadatsch, S., Harrington, R., and Verkerke, W. (2015). Interpolation between multi-dimensional histograms using a new non-linear moment morphing method. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 771:39–48.

Baldi, P., Cranmer, K., Faucett, T., Sadowski, P., and Whiteson, D. (2016). Parameterized Machine Learning for High-Energy Physics.

Bickel, S., Brückner, M., and Scheffer, T. (2009). Discriminative learning under covariate shift. *The Journal of Machine Learning Research*, 10:2137–2155.

Brochu, E., Cora, V. M., and De Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Chen, Y., Di Marco, E., Lykken, J., Spiropulu, M., Vega-Morales, R., et al. (2015). 8D likelihood effective Higgs couplings extraction framework in $h \to 4\ell$. *JHEP*, 1501:125.

Cowan, G., Cranmer, K., Gross, E., and Vitells, O. (2010). Asymptotic formulae for likelihood-based tests of new physics. *Eur.Phys.J.*, C71:1554.

Cranmer, K., Lewis, G., Moneta, L., Shibata, A., and Verkerke, W. (2012). HistFactory: A tool for creating statistical models for use with RooFit and RooStats.

Cranmer, K., Pavez, J., and Louppe, G. (2016). carl: a toolkit for likelihood free inference. `https://github.com/diana-hep/carl`.

Friedman, J., Hastie, T., Tibshirani, R., et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407.

Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schölkopf, B. (2009). Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5.

Hido, S., Tsuboi, Y., Kashima, H., Sugiyama, M., and Kanamori, T. (2011). Statistical outlier detection using direct density ratio estimation. *Knowledge and information systems*, 26(2):309–336.

Hörmander, L. (1990). *The Analysis of Linear Partial Differential Operators I.* Springer Science, Business Media.

Ihler, A., Fisher, J., and Willsky, A. (2004). Nonparametric Hypothesis Tests for Statistical Dependency. *IEEE Transactions on Signal Processing*, 52(8):2234–2249.

Lin, Y. (2002). Support vector machines and the bayes rule in classification. *Data Mining and Knowledge Discovery*, 6(3):259–275.

Neal, R. M. (2007). Computing likelihood functions for high-energy physics experiments when distributions are defined by simulators with nuisance parameters. In *Proceedings of PhyStat2007, CERN-2008-001*, pages 111–118.

Nguyen, X., Wainwright, M. J., Jordan, M., et al. (2010). Estimating divergence functionals and the likelihood ratio by convex risk minimization. *Information Theory, IEEE Transactions on*, 56(11):5847–5861.

Read, A. (1999). Linear interpolation of histograms. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 425(1):357–360.

Scott, C. and Nowak, R. (2005). A neyman-pearson approach to statistical learning. *IEEE Trans. Inform. Theory*, 51:3806–3819.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244.

Sjostrand, T., Mrenna, S., and Skands, P. Z. (2006). PYTHIA 6.4 Physics and Manual. *JHEP*, 0605:026.

Sugiyama, M. and Kawanabe, M. (2012). *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT Press.

Sugiyama, M., Krauledat, M., and Müller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *The Journal of Machine Learning Research*, 8:985–1005.

Sugiyama, M. and Müller, K.-R. (2005). Input-dependent estimation of generalization error under covariate shift. *Statistics & Decisions*, 23(4/2005):249–279.

Sugiyama, M., Suzuki, T., and Kanamori, T. (2012). *Density ratio estimation in machine learning*. Cambridge University Press.

The ATLAS Collaboration (2012). Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. *Phys.Lett.*, B716:1–29.

The CMS Collaboration (2012). Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. *Phys.Lett.*, B716:30–61.

Vapnik, V. (1998). *Statistical learning theory*, volume 1. Wiley New York.

Vapnik, V., Braga, I., and Izmailov, R. (2013). Constructive setting of the density ratio estimation problem and its rigorous solution. *arXiv preprint arXiv:1306.0407*.

Volobouev, I. (2011). Matrix Element Method in HEP: Transfer Functions, Efficiencies, and Likelihood Normalization.

Xin Tong (2013). A Plug-in Approach to Neyman-Pearson Classification. *Journal of Machine Learning Research*, 14:3011–3040.

# A Probabilistic classification for building $s$

In this appendix, we show for completeness that the probabilistic classification framework yields a reduction $s$ which satisfies conditions of Theorem 1.

**Proposition 2.** *Let $\mathbf{X} = (X_1, ..., X_p)$ and $Y$ be random input and output variables with values in $\mathcal{X} \subseteq \mathbb{R}^p$ and $\mathcal{Y} = \{0, 1\}$ and mixed joint probability density function $p_{\mathbf{X},Y}(\mathbf{x}, y)$. For the squared error loss, the best regression function $s : \mathcal{X} \mapsto [0, 1]$, or equivalently the best probabilistic classifier, is*

$$s^*(\mathbf{x}) = \frac{P(Y=1)p_{\mathbf{X}|Y}(\mathbf{x}|Y=1)}{P(Y=0)p_{\mathbf{X}|Y}(\mathbf{x}|Y=0) + P(Y=1)p_{\mathbf{X}|Y}(\mathbf{x}|Y=1)}. \tag{A.1}$$

*Proof.* For the squared error loss,

$$\begin{aligned}
s^*(\mathbf{x}) &= \arg\min_{s(\mathbf{x})} \mathbb{E}_{Y|\mathbf{X}=\mathbf{x}}\{(Y - s(\mathbf{x}))^2\} \\
&= \arg\min_{s(\mathbf{x})} \mathbb{E}_{Y|\mathbf{X}=\mathbf{x}}\{Y^2\} - 2s(\mathbf{x})\mathbb{E}_{Y|\mathbf{X}=\mathbf{x}}\{Y\} + s(\mathbf{x})^2 \\
&= \arg\min_{s(\mathbf{x})} -2s(\mathbf{x})\mathbb{E}_{Y|\mathbf{X}=\mathbf{x}}\{Y\} + s(\mathbf{x})^2 \qquad \text{(A.2)}
\end{aligned}$$

The last expression is minimized when $\frac{d}{ds(\mathbf{x})}(-2s(\mathbf{x})\mathbb{E}_{Y|\mathbf{X}=\mathbf{x}}\{Y\} + s(\mathbf{x})^2) = 0$, that is when $-2\mathbb{E}_{Y|\mathbf{X}=\mathbf{x}}\{Y\} + 2s(\mathbf{x}) = 0$, hence

$$s^*(\mathbf{x}) = \mathbb{E}_{Y|\mathbf{X}=\mathbf{x}}\{Y\}. \qquad \text{(A.3)}$$

For $\mathcal{Y} = \{0, 1\}$,

$$\begin{aligned}
\mathbb{E}_{Y|\mathbf{X}=\mathbf{x}}\{Y\} &= P(Y = 0|\mathbf{X} = \mathbf{x}) \times 0 + P(Y = 1|\mathbf{X} = \mathbf{x}) \times 1 \\
&= \frac{P(Y = 1)p_{\mathbf{X}|Y}(\mathbf{x}|Y = 1)}{p_{\mathbf{X}}(\mathbf{x})} \\
&= \frac{P(Y = 1)p_{\mathbf{X}|Y}(\mathbf{x}|Y = 1)}{P(Y = 0)p_{\mathbf{X}|Y}(\mathbf{x}|Y = 0) + P(Y = 1)p_{\mathbf{X}|Y}(\mathbf{x}|Y = 1)}. \qquad \text{(A.4)}
\end{aligned}$$

□

For $P(Y = 0) = P(Y = 1) = \frac{1}{2}$, the best regression function $s^*$ simplifies to

$$s^*(\mathbf{x}) = \frac{p_{\mathbf{X}|Y}(\mathbf{x}|Y = 1)}{p_{\mathbf{X}|Y}(\mathbf{x}|Y = 0) + p_{\mathbf{X}|Y}(\mathbf{x}|Y = 1)}. \qquad \text{(A.5)}$$

If we further assume that samples for $Y = 0$ (resp. $Y = 1$) are drawn from some parameterized distribution with probability density $p_{\mathbf{X}}(\mathbf{x}|\theta_0)$ (resp. $p_{\mathbf{X}}(\mathbf{x}|\theta_1)$), then the best regression function can be rewritten as

$$s^*(\mathbf{x}) = \frac{p_{\mathbf{X}}(\mathbf{x}|\theta_1)}{p_{\mathbf{X}}(\mathbf{x}|\theta_0) + p_{\mathbf{X}}(\mathbf{x}|\theta_1)}. \qquad \text{(A.6)}$$

In particular, this regression function satisfies conditions of Theorem 1 since $s^*(\mathbf{x}) = m(\frac{p_{\mathbf{X}}(\mathbf{x}|\theta_0)}{p_{\mathbf{X}}(\mathbf{x}|\theta_1)})$, for $m(r(\mathbf{x})) = \frac{1}{1+r(\mathbf{x})}$, is monotonic with $\frac{p_{\mathbf{X}}(\mathbf{x}|\theta_0)}{p_{\mathbf{X}}(\mathbf{x}|\theta_1)}$.

Proposition 2 holds for the squared error loss, but it can be similarly shown that classifiers minimizing the exponential loss, the binomial log-likelihood (or cross-entropy) or the squared hinge loss are also monotonic with the density ratio (Friedman et al., 2000; Lin, 2002). However, a classifier with discrete outputs and minimizing the zero-one loss does not satisfy conditions of the theorem.