Approximating Likelihood Ratios with Calibrated Discriminative Classifiers

Kyle Cranmer, Juan Pavez and Gilles Louppe
New York University

February 10, 2016

Abstract

In many fields of science, generalized likelihood ratio tests are established tools for statistical inference. In practice, these tests are often complicated by the fact that computer simulators are used as a generative model for the data, which does not provide a way to evaluate the likelihood function. In this paper, we demonstrate that likelihood ratios are invariant under dimensionality reductions $\mathbb{R}^p \mapsto \mathbb{R}$, provided the transformation is itself monotonic with the likelihood ratio. 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. In particular, the proposed method offers a machine learning-based approach to statistical inference that is complementary to likelihood-free Bayesian inference algorithms, such as Approximate Bayesian Computation, as it does not require the definition of a prior over model parameters. Experimental results on artificial problems 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 θ 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 particular, to improve the statistical power of these tests, hundreds of these searches have already been using supervised learning to train discriminative classifiers that take advantage of a high dimensional feature vector $\mathbf{x}$.

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 θ. 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 (θ_0,θ_1).

Theorem 1. *Let* $\mathbf{X}$ *be a random vector 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 $r(\mathbf{x};\theta_0,\theta_1) = \frac{p_{\mathbf{X}}(\mathbf{x}|\theta_0)}{p_{\mathbf{X}}(\mathbf{x}|\theta_1)}$*, for given parameters* θ_0 *and* θ_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 δ 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 $\frac{p_{\mathbf{X}}(\mathbf{x}|\theta_0)}{p_{\mathbf{X}}(\mathbf{x}|\theta_1)}$, it exists an invertible function $m : \mathbb{R}^+ \mapsto \mathbb{R}$ such that $s(\mathbf{x}) = m(\frac{p_{\mathbf{X}}(\mathbf{x}|\theta_0)}{p_{\mathbf{X}}(\mathbf{x}|\theta_1)})$. 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 $\frac{p_{\mathbf{X}}(\mathbf{x}|\theta_0)}{p_{\mathbf{X}}(\mathbf{x}|\theta_1)}$ 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 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, i.e., not for $m(r(\mathbf{x})) = \frac{1}{1+r(\mathbf{x})}$ only. Equivalently, Theorem 1 shows that in case 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.

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 particular, in such cases where $s(\mathbf{x})$ is not monotonic with $r(\mathbf{x})$, the induced probability does not factorize as in Eqn. 2.8, resulting in a biased approximation of the density ratio.

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 or calibration algorithms, such as histograms, KDE or isotonic regression, and as applied on $\{\hat{s}(\mathbf{x}_i)\}$, for samples $\{\mathbf{x}_i\}$ 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.

One strength of the proposed approach is that it factorizes the approximation of the per-sample reduction ($\hat{s}(\mathbf{x}) \approx s(\mathbf{x})$) from the calibration procedure ($\hat{p}(\hat{s}(\mathbf{x})|\theta) \approx p(\hat{s}(\mathbf{x})|\theta)$).

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 $r(\mathbf{x}; \theta_0, \theta_1) = \frac{p(\mathbf{x}|\theta_0)}{p(\mathbf{x}|\theta_1)}$ 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 θ_0 and θ_1 to the case of composite hypothesis testing where θ 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

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

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 μ

components are considered parameters of interest while the ν components are considered nuisance parameters. In that case Θ_0 corresponds to all values of ν with μ 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 θ_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 (θ,θ_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 θ.

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 θ_0 and θ_1, the parameters associated to the null and alternate hypotheses, respectively. While this could be done independently for all θ_0 and θ_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 θ_0, θ_1.

While the optimal decision function 2.10 is expected to be learned for the parameter values θ_0 and θ_1 selected in Algorithm 1, it is not clear whether the optimal decision function can be expected for data generated from θ_0' and θ_1' never jointly encountered at learning. Similarly, it is not clear how the limited capacity of the classifier may impact the performance the resulting parameterized decision function. These issues are left as an area for future work, but preliminary exploration by Baldi et al. (2016) show that Algorithm 1 is an effective practical approach.

3.3 Parameterized calibration

Once the classifier is trained, we can use the generative model together with a univariate density estimation technique (e.g. histograms or kernel density estimation) to approximate $p(s|\theta)$ for specific parameter points. For a single parameter point θ, this is a tractable

Algorithm 1 Learning a parameterized classifier.

$\mathcal{L} := \{\}$;

while size($\mathcal{L}$) $< N$ **do**

Select or draw θ_0, θ_1 from Θ_0, Θ_1;

Draw $\mathbf{x} \sim p(\mathbf{x}|\theta_0)$;

$\mathcal{L} := \mathcal{L} \cup \{((\mathbf{x}, \theta_0, \theta_1), y = 0)\}$;

Draw $\mathbf{x} \sim p(\mathbf{x}|\theta_1)$;

$\mathcal{L} := \mathcal{L} \cup \{((\mathbf{x}, \theta_0, \theta_1), y = 1)\}$;

end while

Learn a single classifier $s(\mathbf{x}; \theta_0, \theta_1)$ from $\mathcal{L}$.

univariate density estimation problem. The challenge comes from the need to calibrate this density for all values of θ. A straightforward approach would be to run the generative model on demand for any particular value of θ. In the context of a likelihood fit this would mean that the optimization algorithm that is trying to maximize the likelihood with respect to θ needs access to the generative model $p(\mathbf{x}|\theta)$. This 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, with a fixed classifier, it has become common to interpolate the distribution between discrete values of θ 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 embedding the classifier and estimating the density $p(s|\theta)$ and the relative merits of those approaches will depend critically on the dimensionality of θ 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 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. We indeed have

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

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 forces 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 θ. For this reason, sub-ratios $\frac{p_{c'}(s_{c,c'}(\mathbf{x};\theta_0,\theta_1)|\theta_1)}{p_c(s_{c,c'}(\mathbf{x};\theta_0,\theta_1)|\theta_0)}$ simplify to $\frac{p_{c'}(s_{c,c'}(\mathbf{x}))}{p_c(s_{c,c'}(\mathbf{x}))}$, and can all be pre-computed without having to resort to parameterized classification and calibration.

4 Examples and applications

4.1 Toy example

As a simple but illustrative example, let us first consider the approximation of the likelihood log-ratio $\log\left(r(\mathbf{x};\theta_0,\theta_1)\right)$ between the mixtures $p(\mathbf{x}|\theta_0)$ and $p(\mathbf{x}|\theta_1)$ illustrated in Figure 1a and defined as

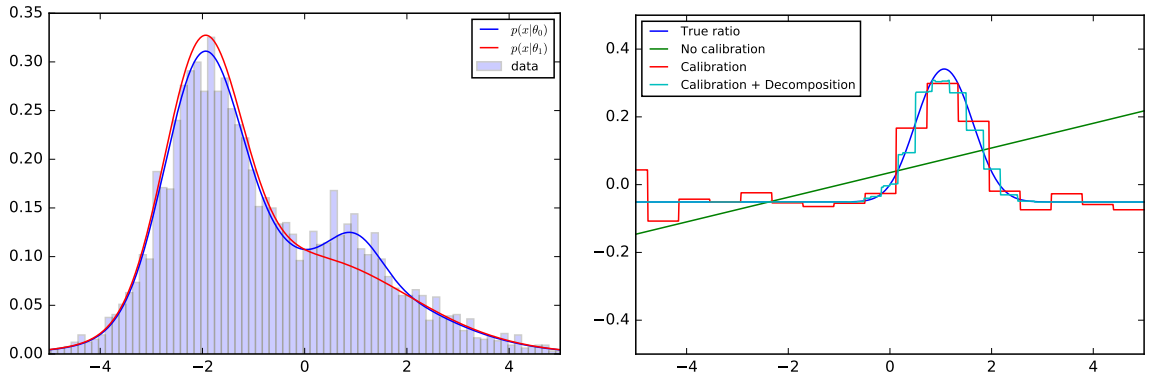
$$p(\mathbf{x}|\theta_0) = (\frac{1}{2} - g)p_{c_0}(\mathbf{x}) + (\frac{1}{2} - g)p_{c_1}(\mathbf{x}) + gp_{c_2}(\mathbf{x}), \tag{4.1}$$

$$p(\mathbf{x}|\theta_1) = \frac{1}{2}p_{c_0}(\mathbf{x}) + \frac{1}{2}p_{c_1}(\mathbf{x}), \tag{4.2}$$

where $p_{c_0} := \mathcal{N}(\mu = -2, \sigma^2 = 0.5625)$, $p_{c_1} := \mathcal{N}(\mu = 0, \sigma^2 = 4)$, $p_{c_2} := \mathcal{N}(\mu = 1, \sigma^2 = 0.25)$ and where g is a parameter of $p(\mathbf{x}|\theta_0)$ set to 0.05. Samples drawn from $p(\mathbf{x}|\theta_0)$ are shown in the figure and will be used later for inference.

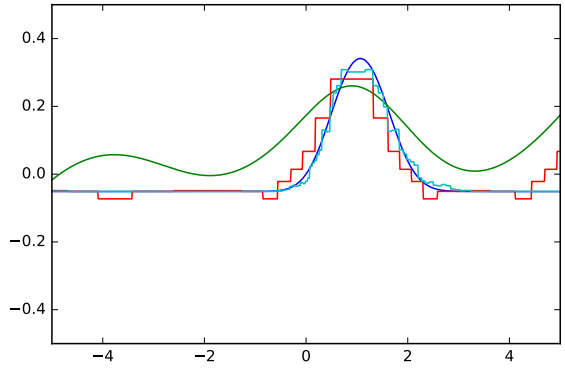
Using the method outlined in Section 2.2, figures 1b, 1c and 1d show the approximated ratio when using respectively a linear model, a 2-layer neural network or a random forest. The blue curve corresponds to the true target log-ratio, the green curve is the direct approximation $\log\left(\frac{\hat{s}(\mathbf{x})}{1-\hat{s}(\mathbf{x})}\right)$ without calibration, while the red curve is the approximation $\log\frac{\hat{p}(\hat{s}(\mathbf{x})|\theta_0)}{\hat{p}(\hat{s}(\mathbf{x})|\theta_1)}$ as calibrated using histograms. Finally, the cyan curve shows the approximated ratio when decomposing the mixtures, as described in Section 3.4. Hyper-parameters of the classifiers are all tuned on a finite grid of values in order to minimize the mean squared error, as estimated by 3-fold cross-validation.

The first fact to highlight is that, in the finite sample setting, uncalibrated direct approximations $\log\left(\frac{\hat{s}(\mathbf{x})}{1-\hat{s}(\mathbf{x})}\right)$, as shown in green, do not yield very accurate estimates of the ratio, even when using universal approximators such as a neural network or a random forest. This observation is at the origin of this work, and led us to show that the ratio $r(\mathbf{x})$ can still be computed exactly by forming the equivalent statistic $\frac{p(s(\mathbf{x})|\theta_0)}{p(s(\mathbf{x})|\theta_1)}$, provided

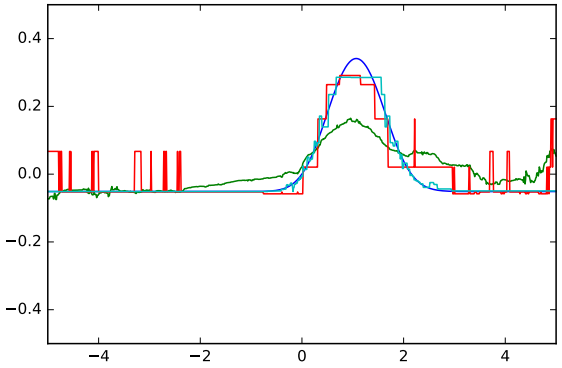


(a) $p(\mathbf{x}|\theta_0)$ and $p(\mathbf{x}|\theta_1)$

(b) Linear model



(c) Neural network



(d) Random forest

Figure 1: Approximation of the log-likelihood ratio $\log r(\mathbf{x}; \theta_0, \theta_1) = \log \frac{p(\mathbf{x}|\theta_0)}{p(\mathbf{x}|\theta_1)}$ with calibrated classifiers.

the transformation $s(\mathbf{x})$ is monotonic with $r(\mathbf{x})$. Indeed, when constructing $\frac{\hat{p}(\hat{s}(\mathbf{x})|\theta_0)}{\hat{p}(\hat{s}(\mathbf{x})|\theta_1)}$ as we propose, estimates appear much closer to the true ratio than their direct counterparts, as illustrated in red. The approximated ratios show to be accurate [GL: Would be nice to be quantitative here.] for all three classifiers, in particular in regions with enough density (i.e., from $\mathbf{x} = -3$ to $\mathbf{x} = 2$) and where the capacity of the classifier is focused. By contrast, in regions where the densities are low (i.e., at the boundaries) the approximated ratio show to have high variance since the underlying classifier cannot make very accurate predictions, because of the few training samples it is trained on in these parts of the input space. Accordingly, when leveraging the fact that densities are mixtures and decomposing their ratio, the capacity of the underlying classifiers can be focused on easier supervised learning tasks, resulting as expected in even more accurate approximations, as shown in cyan for all three classifiers.

As shown in the figures, qualitatively different results are obtained depending the classification algorithm embedded within the approximation of the ratio. In this case, the neural network-based approximation appears to work best, due to the capacity of the classifier to model a decision function $s(\mathbf{x})$ continuous and almost monotonic with the ratio. By contrast, the random forest yields a decision function which is piece-wise constant, therefore non-continuous and not strictly monotonic with the ratio in the finite sample regime, which appears to result in more variance in the approximated ratio. Finally, the linear model produces a decision function that is continuous but only piece-wise monotonic with the ratio. Due to the symmetry of the ratio around $\mathbf{x} = 1$, this still results in an accurate approximation, but poorer results should be expected on more complex problems. Similarly, calibrating $\hat{s}(\mathbf{x})$ to construct $\hat{p}(\hat{s}(\mathbf{x}))$ is also key to obtain accurate results. While standard histograms with fixed bins have been used here for illustrative purposes, we anticipate that the embedded density estimation algorithm should be carefully chosen and

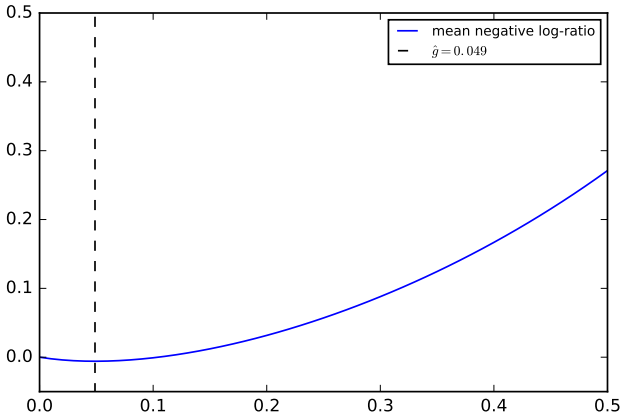


Figure 2: Using approximated likelihood ratios for parameter inference.

tuned, on a case by case basis.

Let us now consider the composite hypothesis testing setting, by comparing $H_0 : g = 0$ versus the alternative $H_1 : g > 0$. That is, we want to evaluate

$$\Lambda(\mathcal{D}; \Theta_0 = \{0\}, \Theta = (0, \frac{1}{2}]) = \frac{p(\mathcal{D}|g = 0)}{\sup_{\theta \in \Theta} p(\mathcal{D}|g = \theta)} \; . \tag{4.3}$$

Following the approach outlined in Section 3, computing the generalized likelihood ratio statistic requires finding the maximum likelihood estimator for the denominator, which can be carried out using approximated likelihood ratios. Reusing the decomposed neural network-based approximation (or equivalently, using a parameterized classifier) and minimizing the mean negative log-ratio of some observed data generated for $g = 0.05$ (as illustrated in Figure 1a), we successfully find as estimator $\hat{g} = 0.049$, as shown in Figure 2. From there, the generalized likelihood ratio can be approximated as in the simple hypothesis setting.

[GL: Smoothly interpolate between histogram bins, in order to get a smooth ratio as well?] [GL: Discuss diagnostics?] [GL: Better highlight advantages of the method?] [GL: Conduct more thorough and convincing experiments?]

4.2 High energy physics

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, and where ν 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 μ is the mixture coefficient associated to the signal component, that is

$$p(\mathcal{D}|\mu,\nu) = \prod_{\mathbf{x}\in\mathcal{D}} \left[\mu p_s(\mathbf{x}|\nu) + (1-\mu)p_b(\mathbf{x}|\nu)\right] \ . \tag{4.4}$$

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.

In this setting, a classifier is usually built from a large sample of data $\{\mathbf{x}_i, y_i\}$ generated with some nominal values of the parameters $\nu = \nu_0$, with $y = 0$ for samples drawn from the background component and $y = 1$ for samples drawn from the signal component. Importantly, the $y = 1$ label corresponds to the signal only hypothesis, and not to the alternate signal-plus-background hypothesis. The resulting classifier approximates the regression function $\frac{p_s(\mathbf{x}|\nu_0)}{p_s(\mathbf{x}|\nu_0)+p_b(\mathbf{x}|\nu_0)}$, which is one to one with the likelihood ratio of the null to the alternate $\frac{p(\mathbf{x}|\mu=0,\nu_0)}{p(\mathbf{x}|\mu,\nu_0)}$ for all μ and therefore satisfies conditions of Theorem 1. Associating the $y = 1$ label to the signal component has advantages because it helps the classifier focus its capacity on the relevant regions in the feature space, particularly when the signal is a very small perturbation to the background (i.e. $\mu \ll 1$). Once the classifier is trained, large samples of data are generated from $p_s(\mathbf{x}|\nu)$ and $p_b(\mathbf{x}|\nu)$ and we estimate the distributions $\hat{p}_s(\hat{s}(\mathbf{x})|\nu)$ and $\hat{p}_b(\hat{s}(\mathbf{x})|\nu)$ continuously parameterized in ν.

Nuisance parameters are an after thought in the typical usage of machine learning in high energy physics. In fact, most discussions related to the training and optimizing the classifier only consider $p_b(\mathbf{x})$ and $p_s(\mathbf{x})$, with $\nu = \nu_0$ being implicit. However, as experimentalists we know that we must account for various forms of systematic uncertainty, parameterized by nuisance parameters ν. In practice, we take the classifier $\hat{s}$ as fixed and then propagate uncertainty by estimating $\hat{p}_s(\hat{s}(\mathbf{x})|\nu)$. Building such distributions for values of ν other than the nominal ν_0 used at training can be thought of as a calibration necessary for classical statistical inference. However, this classifier is clearly not optimal for $\nu \neq \nu_0$. For the this reason, it is expected that the integrated parameterized approach proposed in this work would yield more accurate estimates of the generalized likelihood ratio.

While the original motivation for this work was to improve the treatment of systematic uncertainties in new particle searches by parameterizing the classifier in terms of the nuisance parameters ν, the same approach can be used for parameter inference. In the case of new particle searches, the parameter of interest is the mixture coefficient for the signal component $p_s(x|\nu)$. When measuring particle properties, the distribution of the features also depend on parameters such as a particle's mass and quantum numbers. This is easily accommodated by extending $p_s(\mathbf{x}|\nu) \to p_s(\mathbf{x}|\theta)$, where θ includes both parameters of interest and nuisance parameters. 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. The work of Whiteson and Whiteson (2007) is similar, as the stochastic optimization was directly trying to minimize the measurement uncertainty of a particle's mass; however, the resulting classifier was fixed. This approach also offers the advantage that it explicitly reformulates the per-experiment optimization to the per-event optimization, which is less computationally intensive.

Another approach that is similar in spirit is the so-called matrix element method, in which one directly computes an approximate likelihood ratio by performing a computationally intensive integral associated to the detector response (Volobouev, 2011). In the approach considered in this paper, the detector response is naturally handled by the Monte Carlo sampling used in the simulation of the detector; 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. 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 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 θ with 5-10 parameters of interest and possibly many more nuisance parameters. The state of the art for the operator coefficients of the Higgs decay uses the so-called matrix element likelihood analysis (MELA) in which the equivalent of $s(\mathbf{x}; \theta_0, \theta_1)$ is approximated by neglecting detector effects (Gao et al., 2010; Bolognesi et al., 2012).

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 θ as an input to serve as a dimensionality reduction map. The key distinction comes in the handling of θ. Neal argues that a classifier cannot be used on real data, since we do not know the correct value for θ, 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 θ, 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 θ. 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 θ 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 em-

phasizes 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 probability densities from given and 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 $r(\mathbf{x}; \theta_0, \theta_1) = \frac{p(\mathbf{x}|\theta_0)}{p(\mathbf{x}|\theta_1)}$ 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 θ. Different strategies to this calibration are anticipated depending on the dimensionality of θ, the complexity of the resulting likelihood function, or the practical issues associated to running the simulator.

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.

Bolognesi, S., Gao, Y., Gritsan, A. V., Melnikov, K., Schulze, M., et al. (2012). On the spin and parity of a single-produced resonance at the LHC. *Phys.Rev.*, D86:095031.

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.

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.

Gao, Y., Gritsan, A. V., Guo, Z., Melnikov, K., Schulze, M., et al. (2010). Spin determination of single-produced resonances at hadron colliders. *Phys.Rev.*, D81:075022.

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.

Whiteson, S. and Whiteson, D. (2007). Stochastic optimization for collision selection in high energy physics. *Association for the Advancement of Artificial Intelligence*, 292.

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,

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

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

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

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

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

□

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

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

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

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

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.