Approximating generalized likelihood ratio tests with calibrated discriminative classifiers

Anonymous Author(s)
Affiliation
Address
email

Abstract

We demonstrate how discriminative classifiers can be used to approximate generalized likelihood ratio tests over high-dimensional data when a generative model for the data is available for training and calibration.

1 Introduction

In many areas of science, (generalized) likelihood ratio tests are established tools for statistical inference [1]. It is increasingly common that a generative model is used to describe complex processes that tie parameters θ of an underlying theory and measurement apparatus to high-dimensional observations x. Directly evaluating the likelihood ratio in these cases is often not possible or is computationally impractical. Here we demonstrate how discriminative classifiers can be used to construct equivalent likelihood ratio tests when a generative model for the data is available for training and calibration.

As a concrete example, consider searches for new particles at the Large Hadron Collider. The ATLAS and CMS experiments have published several hundred papers where the final statistical result was formulated as a hypothesis test or confidence interval using a generalized likelihood ratio test [2]. This includes the discovery of the Higgs boson [3, 4] 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*). To improve the statistical power of these tests, hundreds of these searches have utilized supervised learning to train discriminative classifiers. Within High Energy Physics (HEP) libraries such as TMVA that implement conventional techniques like the multi-layer perceptron and boosted decision trees [5] are commonly used. Recently, there has been progress in using deep networks [6] and a NIPS workshop synthesizing the lessons learned during HiggsML [7], the largest Kaggle challenge in history.

While classification accuracy can lead to optimal approaches for simple hypothesis tests [8], that is no longer true in the context of parameter estimation or composite hypothesis tests with nuisance parameters. As noted in [9], "such methods are suboptimal because they assume that the selector with the highest classification accuracy will yield a mass measurement with the smallest statistical uncertainty." A fundamental distinction is that evaluating the loss for classification is a per-event operation, while evaluating the loss for a mass measurement (and other estimation problems) is a per-experiment operation involving a dataset with many events. They went on to demonstrate a computationally intensive stochastic optimization technique based on the per-experiment loss out performed the two stage selection-estimation process.

Generalized likelihood ratio tests enjoy a number of optimal properties in the asymptotic limit and are widely used even away from that limit. These asymptotic properties, the difficulty in specifying

a prior over the parameters needed to evaluate a Bayes risk, and the vast literature associated to likelihood ratio tests has led many users of such tests to not be explicit about the loss function they wish to minimize for their experiment. Thus, we will not try to specify a loss function other than to approximate the generalized likelihood ratio itself. The form of the generalized likelihood ratio is explicitly a per-experiment object, thus naively it has the same computational challenges as other approaches that optimize a per-experiment loss. This paper shows how one can reformulate generalized likelihood ratio test in terms of a per-event classification stage and a calibration process. Moreover, one can utilize a generative statistical model for the data both to train a discriminative classifier with a supervised learning algorithm and for the calibration process.

1.1 Notation and Assumptions

We use the following notation:

- x: a vector of features for an event
- D: a dataset of $D = \{x_1, \ldots, x_n\}$, where x_e are assumed to be i.i.d.
- θ: parameters of a statistical model
- $p(x|\theta)$: probability density (statistical model) for x given θ
- $s(x; \theta_0, \theta_1)$: real-valued discriminative classification score, parametrized by θ_0 and θ_1
- $p(s_{\theta_0,\theta_1}|\theta)$: The probability density for $s(x; \theta_0, \theta_1)$ implied by $p(x|\theta)$

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

1.2 Prelude

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

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

is the most powerful test statistic. In order to evaluate $T(D)$, one must be able to evaluate the probability density $p(x|\theta)$ at any value x. However, it is increasingly common in science that one has a complex simulation that can act as generative model for $p(x|\theta)$, but one cannot evaluate the density directly. For instance, this is the case high energy physics where the simulation of particle detectors can only be done in the 'forward mode'. This same setting has been considered by [10] and [11].

The main result of this paper is that one can form an equivalent test based on

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

if

$$s_e = s(x_e; \theta_0, \theta_1) = m\left(\, p(x_e|\theta_0)/p(x_e|\theta_1) \,\right) \tag{3}$$

where m is any strictly increasing or decreasing function. This result will be proven below. This allows us to recast the original likelihood ratio test into an alternate form in which supervised learning is used to train the discriminative classifier $s(x; \theta_0, \theta_1)$. The discriminative classifier can be trained with data $(x, y = 0)$ generated from $p(x|\theta_0)$ and $(x, y = 1)$ generated from $p(x|\theta_1)$. In Section 3 we extend this result to generalized likelihood ratio tests, where it will be useful to have the classifier parametrized in terms of (θ_0, θ_1).

While the original goal for frequentist hypothesis testing is to make a decision to accept or reject the null hypothesis based on the entire dataset D, we are able to reformulate it such that the machine learning problem is a per-event classification problem. This follows from the fact that we assume the x_e to be i.i.d.

1.3 Comments on classification and frequentist hypothesis tests

Vast literature exists around generative and discriminative classifiers [12]. Typically, generative classifiers learn a model for the joint probability $p(x, y)$, of the inputs x and the classification label y, and predict $p(y|x)$ via Bayes rule. In contrast, discriminative classifiers model the posterior $p(y|x)$ directly. For classification tasks, one then thresholds on $p(y|x)$. In both cases this description in terms of a posterior requires a prior distribution for $p(y)$, which is either modeled explicitly or learned from the training data. This familiar formulation of classification may lead to some confusion in the setting of the current work.

The first possible source of confusion we wish to avoid is that here $p(x|\theta)$ is a generative *statistical model* for the features x, not a generative classifier. We think of the $p(x|\theta)$ along the lines of a traditional scientific theory, able to make predictions about x and being motivated by domain-specific considerations. For example, in the context of HEP $p(x|\theta)$ 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 x. Moreover, we are not attempting to learn the generative model $p(x|\theta)$, we are taking it as given and trying to learn the corresponding (generalized) likelihood ratio test.

The second possible source of confusion is that we are not directly interested in calibrating the classification score in terms of a per-event posterior probability $p(y|x)$. Instead, we are interested in the approximation of the per-experiment likelihood ratio, which can be used to calculate p-values defined by $P(T(D) > k|\theta)$.

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

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

which is monotonic with the desired likelihood ratio $p(x|\theta_1)/p(x|\theta_0)$. Since the prior $p(y)$ is not needed for the target likelihood ratio test and because the classifier score $p(y|x)$ may not be well calibrated, we choose to denote the classifier score $s(x)$ and simply think of it as a deterministic dimensionality reduction map $s : X \to \mathbb{R}$. Similar points have been made by [10].

2 Dimensionality reduction and calibration

We are interested in reformulating the target likelihood ratio

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

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

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

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

leads to an equivalent test.

For notational simplicity, let $p_0(x) = p(x|\theta_0)$, $p_1(x) = p(x|\theta_1)$, and $s(x) = s(x; \theta_1, \theta_0)$. The distribution of x totally determines the distribution of s. In the application at hand, the function

s maps a high-dimensional feature vector x to $\mathbb{R}^+$. Let Ω_c be the level set $\{x \mid s(x) = c\}$ and $\hat{n} = \nabla s(x)/|\nabla s(x)|$ be the orthonormal vector to Ω_c at the point x.

We need to show that for all x, the density

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

is equal to the density

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

It is sufficient to show that $q_x(x) = q_s(s(x))\ \forall x \in \Omega_c$. The function $q_s(s)$ is based on the induced densities $p_0(s)$ and $p_1(s)$. The induced density $p_1(c)$ is given by

$$p_1(c) = \int dx \delta(c - s(x)) p_1(x) = \int d\Omega_c p_1(x)/|\hat{n} \cdot \nabla s| \tag{9}$$

and a similar equation for $p_0(c)$.

Theorem 1: We have the following equality

$$\frac{p_1(c)}{p_0(c)} = \frac{p_1(x)}{p_0(x)} \qquad \forall x \in \Omega_c. \tag{10}$$

Proof For $x \in \Omega_c$, we can factor out of the integral the constant $p_1(x)/p_0(x)$. Thus

$$p_1(c) = \int dx \delta(c - s(x)) p_1(x) = \int d\Omega_c p_1(x)/|\hat{n} \cdot \nabla s| = \frac{p_1(x)}{p_0(x)} \int d\Omega_c p_0(x)/|\hat{n} \cdot \nabla s| \ , \tag{11}$$

and the integrals cancel in the likelihood ratio

$$\frac{p_1(c)}{p_0(c)} = \frac{p_1(x)}{p_0(x)} \frac{\int d\Omega_c p_0(x)/|\hat{n} \cdot \nabla s|}{\int d\Omega_c p_0(x)/|\hat{n} \cdot \nabla s|} = \frac{p_1(x)}{p_0(x)} \qquad \forall x \in \Omega_c. \tag{12}$$

One can think of the ratio $p_1(s)/p_0(s)$ as a way of calibrating the the discriminative classifier and correcting for the monotonic transformation m of the desired likelihood ratio as in Eq. 3.

3 Composite hypotheses and the generalized likelihood ratio

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

$$\lambda(x) = \frac{\sup_{\theta \in \Theta_0} p(D|\theta)}{\sup_{\theta \in \Theta} p(D|\theta)} \ . \tag{13}$$

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.

Denote the maximum likelihood estimator

$$\hat{\theta} = \arg\max_{\theta} p(D|\theta) \tag{14}$$

and the conditional maximum likelihood estimator

$$\hat{\hat{\theta}} = \arg\max_{\theta \in \Theta_0} p(D|\theta) \ . \tag{15}$$

[1] Also known as the profile likelihood ratio.

It is not obvious that if we are working with the distributions $p(s|\theta)$ (for some particular $s(x;\theta_0,\theta_1)$ comparison) that we can find the same estimators. Fortunately, there is a construction based on $p(s|\theta)$ that works. The maximum likelihood estimate of Eq. 14 is the same as the value that maximizes the likelihood ratio with respect to $p(D|\theta_1)$ for some fixed value of θ_1. This allows us to use Theorem 1 to reformulate the maximum likelihood estimate

$$\hat{\theta} = \arg\max_{\theta} \frac{p(D|\theta)}{p(D|\theta_1)} = \arg\max_{\theta} \sum \ln \frac{p(x_e|\theta)}{p(x_e|\theta_1)} = \arg\max_{\theta} \sum \ln \frac{p(s(x_e;\theta,\theta_1)|\theta)}{p(s(x_e;\theta,\theta_1)|\theta_1)} \; . \quad (16)$$

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

4 Learning the correct mapping and its distribution

Thus far we have shown that likelihood ratio tests based on $p(x|\theta_0)/p(x|\theta_1)$ with high dimensional features x can be reproduced via hypothesis tests based on the univariate densities $p(s|\theta)$ for the particular dimensionality reduction map $s(x|\theta_0,\theta_1)$. In order for this approach to be useful in a setting where it is not possible to evaluate the density $p(x|\theta)$, we need to be able to approximate both $s(x|\theta_0,\theta_1)$ and $p(s|\theta)$ based on samples $\{x_i\}$ drawn from the generative model $p(x|\theta)$.

Denote the approximate dimensionality reduction map $\hat{s}(x;\theta_0,\theta_1)$ and its distribution $\hat{g}(\hat{s}|\theta)$. In general we will be interested in the machine learning problem that approximates these distributions based on samples $\{x_i\}$ drawn from the generative model $p(x|\theta)$. The first step in this direction is to confirm that a discriminative classifier obtained from common supervised learning algorithms will yield a function that is one-to-one with $p(x|\theta_0)/p(x|\theta_1)$.

4.1 The standard discriminative classification setting

For fixed θ_0 and θ_1 we can generate large samples from each model and train a classifier. To be concrete, let us use $p(x|\theta_0)$ to generate training data $(x_i, y_i = 0)$ and $p(x|\theta_1)$ to generate training data $(x_i, y_i = 1)$. With balanced training data ($p(y = 1) = p(y = 0) = 1/2$) a quadratic loss function will lead to classifiers that approximate the regression function $\hat{s}(x) \approx p(y|x) = p(x|\theta_1)/(p(x|\theta_0) + p(x|\theta_1))$, which is monotonic with the desired per-event likelihood ratio $q(x)$. Thus, standard supervised learning algorithms with various surrogate loss functions lead to discriminative classifiers that approximate a monotonic function of per-event likelihood ratio $q(x)$. 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 $\hat{g}(\hat{s}|\theta)$.

In the limit of large samples from the generative model, we can approximate the original likelihood ratio test arbitrarily well. With finite training data for $\hat{s}(x)$ and samples to approximate $\hat{g}(\hat{s}|\theta)$ it will be necessary to be more specific about the what loss function we are interested in for approximating the likelihood ratio test. This will depend in general on the ultimate goal of the test. We know that in the case of composite hypothesis tests that there is in general no uniformly most powerful test, thus it is likely that a decision theoretic approach taking into account some weighting or utility over the space Θ is necessary. This is left as a subject for future work.

4.2 Training a parametrized classifier

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

While the function $p(x|\theta_1)/(p(x|\theta_0) + p(x|\theta_1))$ will minimize the expected squared loss based on training data produced according to Algorithm 1, it is not clear how training data from $\theta_0' \neq \theta_0$ and

Algorithm 1 Training of the parametrized classifier.

```
initialize trainingData = {}
for θ0 in Θ do
  for θ1 in Θ do
    generate x_i^0 ~ p(x|θ0)
    append {(x_i^0, θ0, θ1, y = 0)} to trainingData
    generate x_i^1 ~ p(x|θ1)
    append {(x_i^1, θ0, θ1, y = 1)} to trainingData
  end for
end for
use trainingData to learn ŝ(x; θ0, θ1)
```

$\theta_1' \neq \theta_1$ will influence a real world classifier with finite capacity. This is left as an area for future work.

4.3 Embedding the classifier into the likelihood

In most settings that make use of likelihood ratio tests, the likelihood is based directly on some approximation of density for the observed data via $\hat{f}(x|\theta)$. In general, approximating the density $\hat{f}(x|\theta)$ is difficult for high-dimensional data, which motivates the use of the dimensionality reduction map $\hat{s}(x)$ and likelihood ratio tests based on the univariate density $\hat{g}(\hat{s}|\theta)$. In the case of a fixed classifier $\hat{s}(x)$ it is possible to pre-compute $\hat{s}_e = \hat{s}(x_e)$ and never refer back to the original features x_e. In the parametrized setting it is not possible to pre-compute $\hat{s}(x_e; \theta_0, \theta_1)$ for all values of θ_0 and θ_1; however, we can embed the classifier into the likelihood function to carry out the composition $\hat{g} \circ \hat{s}$. A concrete realization of this has been performed for probability models implemented with the `RooFit` probabilistic programing language and classifiers implemented with `scikit-learn` and `TMVA` [13, 14, 5].

In both cases, constructing the density $\hat{p}(\hat{s}|\theta)$ requires running the generative model at 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 $p(x|\theta)$. 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 our HEP examples we have chosen to interpolate the distribution between discrete values of θ in order to produce a continuous parametrization for $\hat{p}(\hat{s}|\theta)$. One can easily imagine a number of approaches to embedding the classifier and estimating the density $\hat{p}(\hat{s}|\theta)$ 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.

5 Applications in HEP

In HEP 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 x that have corresponding distributions $p_b(x|\nu)$ for background and $p_s(x|\nu)$ for signal, 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 associate to the signal component.

$$p(D \,|\, \mu, \nu) = \prod_{e=1}^{n} \left[\, \mu p_s(x_e \,|\, \nu) + (1 - \mu)\, p_b(x_e \,|\, \nu) \, \right] \,, \tag{17}$$

New particle searches at the LHC are typically framed as hypothesis test where the null corresponds to $\mu = 0$, and the generalized likelihood ratio (profiling over ν as in Eq. 15) is used as a test statistic [2]. This approach was used for the discovery of the Higgs boson [3, 4].

5.1 Typical usage pattern

In this setting, large samples of synthetic data $\{x_i, y_i\}$ generated with some nominal values of the parameters ν_0, where $y = 0$ corresponds to the background density $p_b(x|\nu_0)$ and $y = 1$ corresponds to signal density $p_s(x|\nu_0)$. Importantly, the $y = 1$ label corresponds to the signal only, and not to the alternate signal-plus-background hypothesis. The resulting classifier approximates the regression function $p_s(x|\nu_0)/(p_s(x|\nu_0) + p_b(x|\nu_0))$, which is one to one with the likelihood ratio of the null to the alternate $p(x|\mu = 0, \nu_0)/p(x|\mu, \nu_0)$ for all μ. 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.

Once the classifier is trained, large samples of synthetic data drawn from $p_s(x|\nu)$ and $p_b(x|\nu)$, then we estimate the distributions $\hat{p}_s(\hat{s}|\nu)$ and $\hat{p}_b(\hat{s}|\nu)$ continuously parametrized in ν. An example of the distributions of the distribution of $\hat{s}$ for the signal and background events with $\nu = \nu_0$ is shown in Figure **??**.

These steps feed into a subsequent statistical test based on the observed data $D = (x_1, \ldots, x_n)$. For each event, the classifier is evaluated and one performs inference on a parameter μ related to the presence of the signal contribution. In particular, one forms the statistical model[2]

$$p(D \,|\, \mu, \nu) = \prod_{e=1}^{n} \left[\, \mu \hat{p}_s(\hat{s}(x_e) \,|\, \nu) + (1 - \mu)\, \hat{p}_b(\hat{s}(x_e) \,|\, \nu) \, \right] \, . \tag{18}$$

5.2 Comments on typical usage of machine learning in HEP

Nuisance parameters are an after thought in the typical usage of machine learning in HEP. In fact, most discussions would related to the training and optimizing the classifier only consider $p_b(x)$ and $p_s(x)$ with $\nu = \nu_0$ being implicit. However, as experimentalists we know that we must account for various forms of systematic uncertainty, parametrized by nuisance parameters ν. In practice, we take the classifier as fixed and then propagate uncertainty through the classifier as in Eq. 18. Building the distribution $p(\hat{s}|\nu)$ for values of ν other than the nominal ν_0 used to train the classifier can be thought of as a calibration necessary for classical statistical inference; however, this classifier is clearly not optimal for $\nu \neq \nu_0$.

5.3 A more powerful approach

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

$$\lambda(\mu) = \frac{p(D|\mu, \hat{\hat{\nu}})}{p(D|\hat{\mu}, \hat{\nu})} \, , \tag{19}$$

following the approach outlined in Section 3.

There is an interesting distinction between this approach and the standard use in which the classifier is trained for a fixed ν_0. In the standard use one trains a classifier for signal vs. background, which is equivalent (in an ideal setting) to training a classifier for null (background-only) vs. alternate (signal-plus-backgound) since the resulting regression functions are one-to-one with each other. In contrast, in the case of the generalized likelihood ratio test

$$\frac{p(x|0, \hat{\hat{\nu}})}{p(x|\hat{\mu}, \hat{\nu})} = \frac{p_b(x|\hat{\hat{\nu}})}{\hat{\mu} p_s(x_e \,|\, \hat{\nu}) + (1 - \hat{\mu})\, p_b(x_e \,|\, \hat{\nu})} \, , \tag{20}$$

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

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

5.4 Measuring particle properties

While the original motivation for this work was to improve the treatment of systematic uncertainties in new particle searches by parametrizing the classifier in terms of the nuisance parameters ν, the same approach can be used to for parameters of interest. 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 of interest, such as a particle's mass and quantum numbers. This is easily accommodated by extending $p_s(x|\nu) \to p_s(x|\theta)$, where θ includes both parameters of interest and nuisance parameters.

This formalism represents is a significant step forward in the usage of machine learning in HEP, where classifiers have always been used between two static classes of events and not parametrized explicitly in terms of the physical quantities we wish to measure. The work of [9] is the closest example 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 [**?**]. 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 per event to calculate the likelihood ratio $q(x)$. The work here can be seen as aiming at the same conceptual target, but utilizing 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 [**?**]).

Contemporary examples where the technique presented here could have major impact on HEP include the measurement of coefficients to quantum mechanical operators describing the decay of the Higgs boson [**?**] and, if we are so lucky, measurement of the mass of supersymmetric particles in cascade decays [**?**]. Both of these examples involve datasets with many events, each with a feature vector 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 $s(x; \theta_0, \theta_1)$ is approximated by neglecting detector effects [**?**].

5.5 Related work

[10] and [11] consider the machine learning problem associated to Neyman-Pearson hypothesis testing. As in this work, 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. Perhaps a formal treatment similar to the Neyman-Pearson case can be brought to bear in this more general setting. [15] explore a way of leveraging generative models to derive kernel functions for use in discriminative methods. This interesting work is distinct from the point made here in which the generative model is being used for the purpose of providing training data and calibration. [16] consider a hybrid generative/discriminative classifier; however, the goal of that work is not to leverage a generative model for the data, but to use both approaches to learn different subsets of the parameters in a single hybrid classifier. [17] emphasize the importance of calibrated probability estimates from decision trees and naive Bayesian classifiers and investigate various approaches to achieve this. In contrast to that work, we are not interested in calibrated probability estimates for $p(y|x)$ for individual events, but instead we use the calibration to correct for non-linear transformations of the target likelihood ratio and, perhaps, to provide calibrated p-values based on those likelihood ratio tests. [18] 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.

6 Conclusions

We have outlined a technique to reformulate generalized likelihood ratio test over a high-dimensional dataset with multiple events in terms of a univariate density of a classifier score. We have shown that a parametrized family of discriminative classifiers $s(x; \theta_0, \theta_1)$ trained and calibrated with a generative model $p(x|\theta)$ can be used to approximate the likelihood ratio $\prod_i p(x_i|\theta_0)/p(x_i|\theta_1)$ even when it is not possible to evaluate the densities $p(x|\theta)$ for an arbitrary x. Furthermore, this allows one to turn the optimization of the procedure for the per-experiment test into optimization of a per-event classification problem and a calibration procedure. This approach leverages the power of machine learning in a classical statistical setting. In particular, it extends the common usage of machine learning for simple hypothesis testing to include the parameter estimation, confidence intervals, and composite hypothesis testing.

References

[1] S. S. Wilks. The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses. *The Annals of Mathematical Statistics*, 9(1):60–62, March 1938.

[2] Glen Cowan, Kyle Cranmer, Eilam Gross, and Ofer Vitells. Asymptotic formulae for likelihood-based tests of new physics. *Eur.Phys.J.*, C71:1554, July 2010.

[3] The ATLAS Collaboration. 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, 2012.

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

[5] Andreas Hocker, J. Stelzer, F. Tegenfeldt, H. Voss, K. Voss, et al. TMVA - Toolkit for Multivariate Data Analysis. *PoS*, ACAT:040, 2007.

[6] Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for Exotic Particles in High-Energy Physics with Deep Learning. *Nature Commun.*, 5:4308, 2014.

[7] Balázs Kégl, David Rousseau, Glen Cowan, Cécile Germain, and Isabelle Guyon. Hepml workshop `https://sites.google.com/site/hepml14/home`.

[8] A. P. Dempster and M. Schatzoff. Expected significance level as a sensitivity index for test statistics. *Journal of the American Statistical Association*, 60(310):pp. 420–436, 1965.

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

[10] Robert Nowak Clayton Scott. A Neyman-Pearson approach to statistical learning.

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

[12] Michael I. Jordan Andrew Y. Ng. On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes.

[13] Wouter Verkerke and David P. Kirkby. The RooFit toolkit for data modeling. *eConf*, C0303241:MOLT007, 2003.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[15] David Haussler Tommi Jaakkola. Exploiting Generative Models in Discriminative Classifiers.

[16] Andrew Y. Ng Rajat Raina, Yirong Shen and Andrew McCallum. Classification with hybrid generative/discriminative models. 2003.

[17] Charles Elkan Bianca Zadrozny. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers.

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