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

Kyle Cranmer

November 19, 2014

1 Introduction

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

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

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

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

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

is the most powerful test statistic. In order to evaluate $T(D)$, one must be able to evaluate the probability density $f(x|\theta)$ at any value x. However, it is increasingly common in science that one has a complex simulation that can act as generative model for $f(x|\theta)$, 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'.

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

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

if

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

or some monotonic function of that ratio. This will be proven below. This allows us to recast the original likelihood ratio test into an alternate form in which a discriminative classifier is used to learn $s(x; \theta_0, \theta_1)$. The discriminative classifier can be trained with data $(x, y = 0)$ generated from $f(x|\theta_0)$ and $(x, y = 1)$ generated from $f(x|\theta_1)$. In Section 3 we extend this result to generalized likelihood ratio tests, where it will be useful to have the discriminative classifier explicitly 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, the machine learning problem is an event-by-event classification problem. This follows from the fact that we assume the x_e to be i.i.d.

2 Dimensionality reduction

The target hypothesis test is based on

$$\ln T = \sum_{e=1}^{n} \underbrace{\log\left[\frac{f(x_e|\theta_0)}{f(x_e|\theta_1)}\right]}_{q(x_e)} . \tag{4}$$

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

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

$$\ln T' = \sum_{e=1}^{n} \underbrace{\log\left[\frac{g(s_e|\theta_0)}{g(s_e|\theta_1)}\right]}_{q(s_e)} , \tag{5}$$

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

$$f(q_x|\theta) = \int dx \delta(q_x - q_x(x)) f(x|\theta) / |\hat{n} \cdot \nabla q_x| \tag{6}$$

is the same as

$$f(q_s|\theta) = \int dx \delta(q_s - q_s(s(x)))\, f(x|\theta)\, / |\hat{n} \cdot \nabla q_s|\ . \tag{7}$$

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

For notational simplicity, let $f_0(x) = f(x|\theta_0)$, $f_1(x) = f(x|\theta_1)$, and $s(x) = s(x; \theta_1, \theta_0)$. The distribution of x totally determines the distribution of s via the change of variables $x \to s$. 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. The induced density $g_1(c)$ is given by

$$g_1(c) = \int dx \delta(c - s(x)) f_1(x) = \int d\Omega_c f_1(x)/|\hat{n} \cdot \nabla s| \tag{8}$$

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

Theorem 1: We have the following equalities

$$\frac{g_1(c)}{g_0(c)} = s(x) = \frac{f_1(x)}{f_0(x)} \qquad \forall x \in \Omega_c. \tag{9}$$

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

$$g_1(c) = \int dx \delta(c - s(x)) f_1(x) = \int d\Omega_c f_1(x)/|\hat{n} \cdot \nabla s| = s(x) \int d\Omega_c f_0(x)/|\hat{n} \cdot \nabla s|\ , \tag{10}$$

and the integrals cancel in the likelihood ratio

$$\frac{g_1(c)}{g_0(c)} = \frac{s(x) \int d\Omega_c f_0(x)/|\hat{n} \cdot \nabla s|}{\int d\Omega_c f_0(x)/|\hat{n} \cdot \nabla s|} = s(x) = \frac{f_1(x)}{f_0(x)} \qquad \forall x \in \Omega_c. \tag{11}$$

In the case of simple hypothesis testing, θ_0 and θ_1 are specified and there is a unique map $s(x) = s(x_e; \theta_0, \theta_1)$. In that case, the equivalent likelihood ratio test can be performed by first transforming the data to $D_s = \{s_1, \ldots, s_e\}$, constructing the likelihoods

$$g(D_s \,|\, \theta) = \prod_{e=1}^{n} g(s_e \,|\, \theta) \tag{12}$$

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

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

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

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

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

and the conditional maximum likelihood estimator

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

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

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

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

4 Learning the correct mapping and its distribution

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

[1] Also known as the profile likelihood ratio.

evaluating the density $f(x|\theta)$. In order for this approach to be useful, we need to be able to approximate both based on samples $\{(x,\theta)\}$ drawn from the generative model $f(x|\theta)$.

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 $f(x|\theta)$. In particular, is there a loss function such that the function $\hat{s}(x)$ that minimizes the expected loss leads to a function that is one-to-one with $f(x|\theta_0)/f(x|\theta_1)$?

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's use $f(x|\theta_0)$ to generate training data $(x_i, y_i = 0)$ and $f(x|\theta_1)$ to generate training data $(x_i, y_i = 1)$. If we use the squared-loss function, then the expected loss is

$$\mathbb{E}[L] = \int dx f(x|\theta_0)(\hat{s}(x))^2 + \int dx f(x|\theta_1)(1-\hat{s}(x))^2 \; . \tag{17}$$

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

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

$$\Delta = \mathbb{E}[L_{\hat{s}'}] - \mathbb{E}[L_{\hat{s}}] = \int dx f(x|\theta_0)(h^2(x) + 2h(x)\hat{s}(x)) + f(x|\theta_1)(h^2(x) - 2h(x) + 2h(x)\hat{s}(x) \; . \tag{18}$$

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

$$\Delta = \int dx (f(x|\theta_0) + f(x|\theta_1))h^2(x) > 0 \; . \tag{19}$$

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

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

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

This treatment shows that in the asymptotic limit of large samples from the generative model, that we can approximate arbitrarily well the original likelihood ratio test. With finite training data for $\hat{s}(x)$ and samples to approximate $\hat{g}(x)$ 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, not 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, discriminative 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, as latent variables whose values will be specified via the enveloping (generalized) likelihood ratio test. We denote the learned family of classifiers $\hat{s}(x; \theta_0, \theta_0)$, and anticipate the training based roughly on the following algorithmic flow.

Algorithm 1 Training the parametrized classifier

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

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

4.3 Embedding the classifier into the likelihood

In most settings that make use of likelihood ratio tests, the probability densities directly model the observed data via $f(x|\theta)$. In the case of a fixed classifier $\hat{s}(x)$ it is possible to pre-compute $\hat{s}_e = \hat{s}(x_e)$. In the parametrized setting this it is not possible to pre-compute $\hat{s}(x; \theta_0, \theta_1)$ for all values of θ_0 and θ_1, so we must computationally we must embed the classifier into the likelihood function to carry out the composition $g \circ s$. A concrete realization of this has been performed for probability models implemented with the `RooFit` probabilistic programing language and discriminative classifiers implemented with `scikit-learn` and `TMVA`.

In both cases, constructing the density $\hat{g}(\hat{s}|\theta)$ requires running the generative model at θ, and in the context of a likelihood fit this would mean that the optimization algorithm that is trying to maximize θ needs access to the generative model $f(x|\theta)$. 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 practice, one may want to interpolate the distribution between discrete values of θ to produce a continuous parametrization for $\hat{g}(s|\theta)$. In such cases, the properties of the interpolation algorithm should be part of the considerations of the over-arching optimization problem.

5 Typical usage of machine learning in HEP

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

$$f(D \,|\, \mu, \nu) = \prod_{e=1}^{n} \left[\mu f_s(x_e \,|\, \nu) + (1-\mu)\, f_b(x_e \,|\, \nu) \right] \ , \tag{20}$$

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

Often machine learning classification algorithms are trained on large samples of synthetic data $\{x_i, y_i\}$ generated with some nominal values of the parameters ν_0, where $y = 0$ corresponds to background and $y = 1$ corresponds to signal. Following the result of Section 4, the resulting classifier approximates the likelihood ratio of the mixture components $f_s(x|\nu_0)/(f_s(x|\nu_0) + f_b(x|\nu_0))$, which is one to one with the likelihood ratio of the null to the alternate $f(x|\mu = 0, \nu_0)/f(x|\mu, \nu_0)$ for all μ. The resulting classifier is denoted $\hat{s}(x)$. Based on this classifier and large samples of synthetic data drawn from $f_s(x|\nu)$ and $f_b(x|\nu)$ we construct the distributions $g_s(\hat{s}|\nu)$ and $g_b(\hat{s}|\nu)$. 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 1.

These steps lead to a subsequent statistical analysis where one observes in 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

$$g(D \,|\, \mu, \nu) = \prod_{e=1}^{n} \left[\mu g_s(\hat{s}(x_e) \,|\, \nu) + (1-\mu)\, g_b(\hat{s}(x_e) \,|\, \nu) \right] \ , \tag{21}$$

where $\mu = 0$ is the null (background-only) hypothesis and $\mu > 0$ is the alternate (signal-

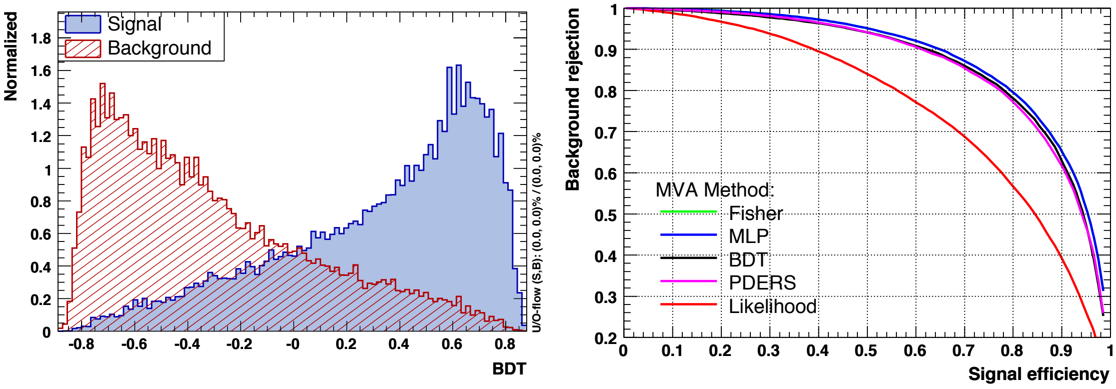

Figure 1: Left: an example of the distributions $g_b(\hat{s}|\nu)$ and $g_s(\hat{s}|\nu)$ when the classifier s is a boosted-decision tree (BDT). Right: the corresponding ROC curve (right) for this and other classifiers. (Figures taken from TMVA manual.)

plus-background) hypothesis.[2] Typically, we are interested in inference on μ and ν are nuisance parameters.

5.1 Comments on typical usage of machine learning in HEP

Nuisance parameters are an after thought in the typical usage of machine learning in HEP. In fact, most machine learning discussions would only consider $f_b(x)$ and $f_s(x)$. However, as experimentalists we know that we must account for various forms of systematic uncertainty, parametrized by nuisance parameters ν. In practice, we take the classifier as fixed and then propagate uncertainty through the classifier as in Eq. 21. Building the distribution $g(\hat{s}|\nu)$ 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 $\nu \neq \nu_0$.

5.2 A more powerful approach

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

$$\lambda(\mu) = \frac{f(D|\mu, \hat{\hat{\nu}})}{f(D|\hat{\mu}, \hat{\nu})} \ , \tag{22}$$

following the approach outlined in Section 3.

[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.

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) as

$$\frac{f(x|0,\nu_0)}{f(x|\hat{\mu},\nu_0)} = \frac{f_b(x|\nu_0)}{\mu f_s(x_e\,|\,\nu_0) + (1-\mu)\,f_b(x_e\,|\,\nu_0)} = \left[c_1 + c_2 \frac{f_s(x|\nu_0)}{f_b(x_e\,|\,\nu_0)}\right]^{-1} \quad , \tag{23}$$

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

$$\frac{f(x|0,\hat{\hat{\nu}})}{f(x|\hat{\mu},\hat{\nu})} = \frac{f_b(x|\hat{\hat{\nu}})}{\hat{\mu} f_s(x_e\,|\,\hat{\nu}) + (1-\hat{\mu})\,f_b(x_e\,|\,\hat{\nu})} \ , \tag{24}$$

the background components don't cancel and there is an additional term $f_b(x|\hat{\hat{\nu}})/f_b(x|\hat{\nu})$. 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.

6 Conclusions

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

Acknowldgements

KC would like to thank Daniel Whiteson for discussions and encouragement throughout the project, Alex Ihler for challenging discussions that led to a reformulation of the initial idea, and Shimon Whiteson for patient feedback in that process. KC would also like to thank Yann LeCun, Philip Stark, and Pierre Baldi for their feedback on the project early in its conception, Balázs Kégl for discussions about the Kaggle challenge and non-standard loss functions, and Yuri Shirman for reassuring discussions on elementary vector calculus. KC is supported by the US National Science Foundation grants PHY-0854724 and PHY-0955626. 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.