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

Kyle Cranmer KYLE.CRANMER@NYU.EDU
Center for Cosmology and Particle Physics
Center for Data Science
New York University
New York, NY 10003, USA

Editor:

Abstract

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, 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 given θ
- $s(x; \theta_0, \theta_1)$: 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 $f(x|\theta)$

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

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

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 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 4 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, we are able to reformulate it such that 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. Comments on classification and frequentist hypothesis tests

Significant literature exists around generative and discriminative classifiers (Andrew Y. Ng). 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 $f(x|\theta)$ is a generative *statistical model* for the features x, not a generative classifier. We think of the $f(x|\theta)$ 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 high energy particle physics $f(x|\theta)$ is based on quantum field theory and 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 $f(x|\theta)$, we are taking it as given and trying to learn the corresponding likelihood ratio test.

The second possible source of confusion is that the likelihood ratio $T(D)$ is aimed at tests based on the entire dataset; we are not interested in thresholded classification on individual events x_e. Additionally, we know that both discriminative and generative classifier scores are often poorly calibrated. For instance, often we wish to have well calibrated p-values defined by $P(T(D) > k|\theta)$, not well calibrated posterior probabilities $p(y|x)$.

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 $f(x|\theta_0)$ and $(x, y = 1)$ generated from $f(x|\theta_1)$, the relative mix $p(y)$ is arbitrary. When $p(y = 0) = p(y = 1) = 1/2$, then

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

which is monotonic with the desired likelihood ratio $f(x|\theta_1)/f(x|\theta_0)$. 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 $s : X \to \mathbb{R}$.

3. Dimensionality reduction and calibration

The target hypothesis test is based on

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

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

The important part of the per-event function $q(x)$ is that it defines iso-contours in the feature space x. As we will show, our goal is to learn a monotonic function of $f(x|\theta_0)/f(x|\theta_1)$, which will share 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 the density $g(s|\theta)$ and that

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

leads to an equivalent test.

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. 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 the density

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

is the same as

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

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 $g_0(s)$ and $g_1(s)$. The induced density $g_1(c)$ is given by

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

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

Theorem 1: We have the following equality

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

Proof For $x \in \Omega_c$, we can factor out of the integral the constant $f_1(x)/f_0(x)$. Thus

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

and the integrals cancel in the likelihood ratio

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

One can think of the ratio $g_1(s)/g_0(s)$ 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.

4. Composite hypotheses and the generalized likelihood ratio

In the case of composite hypotheses $\theta \in \Theta_0$ against an alternative $\theta \in \Theta_0^C$, the generalized likelihood ratio[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 of Eq. 14 is the same as the value that maximizes the likelihood ratio with respect to $f(D|\theta_1)$ for some fixed value of θ_1. This allows us to use Theorem 1 to reformulate the maximum likelihood estimate

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

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

5. Learning the correct mapping and its distribution

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

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 $f(x|\theta)$. The first step in this direction is to confirm that a discriminative classifier obtained from common training procedures will yield a function that is one-to-one with $f(x|\theta_0)/f(x|\theta_1)$.

5.1 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 $(x_i, y_i = 1)$. With balanced training data – p(y=1)=p(y=0)=½ – a quadratic loss function will lead to classifiers that approximate the regression function $\hat{s}(x) \approx p(y|x) = f(x|\theta_1)/(f(x|\theta_0)+f(x|\theta_1))$, which is monotonic with the desired likelihood ratio $f(x|\theta_0)/f(x|\theta_1)$. Thus, standard classification approaches will lead to discriminative classifiers 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 $\hat{g}(\hat{s}|\theta)$.

Thus, in the limit of large samples from the generative model, we can approximate arbitrarily well the original likelihood ratio test. 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.

5.2 Training a parametrized, discriminative classifier

We are left with the practical question of how to train a family of discriminative classifiers parametrized by 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.

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 Algorithm 1, it is not clear how training data

Algorithm 1 Training of the parametrized classifier.

```
initialize trainingData = {}
for θ_0 in Θ do
  for θ_1 in Θ do
    generate x_i^0 ~ f(x|θ_0)
    append {(x_i^0, θ_0, θ_1, y = 0)} to trainingData
    generate x_i^1 ~ f(x|θ_1)
    append {(x_i^1, θ_0, θ_1, y = 1)} to trainingData
  end for
end for
use trainingData to learn ŝ(x; θ_0, θ_1)
```

from $\theta_0' \neq \theta_0$ and $\theta_1' \neq \theta_1$ will influence a real world classifier with finite capacity. This is left as an area for future work.

5.3 Embedding the classifier into the likelihood

In most settings that make use of likelihood ratio tests, the likelihood is based directly on some approximation of density for the observed data via $\hat{f}(x|\theta)$. 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 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 this it is not possible to pre-compute $\hat{s}(x_e; \theta_0, \theta_1)$ for all values of θ_0 and θ_1, so we must 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 discriminative classifiers implemented with `scikit-learn` and `TMVA` (Verkerke and Kirkby, 2003; Pedregosa et al., 2011; Hocker et al., 2007).

In both cases, constructing the density $\hat{g}(\hat{s}|\theta)$ 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 $f(x|\theta)$. 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 $\hat{g}(\hat{s}|\theta)$. In such cases, the properties of the interpolation algorithm should be part of the considerations of the over-arching optimization problem.

6. Typical usage of machine learning in HEP

In high-energy physics (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. Generalized likelihood ratio tests are used widely in HEP (Cowan et al., 2010), most notably in the discovery of the Higgs boson (The ATLAS Collaboration, 2012; The CMS Collaboration, 2012). 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

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

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 the background density $f_b(x|\nu_0)$ and $y = 1$ corresponds to signal density $f_s(x|\nu_0)$ (not the signal-plus-background). The resulting classifier approximates the regression function $f_s(x|\nu_0)/(f_s(x|\nu_0) + f_b(x|\nu_0))$, 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

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

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

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.

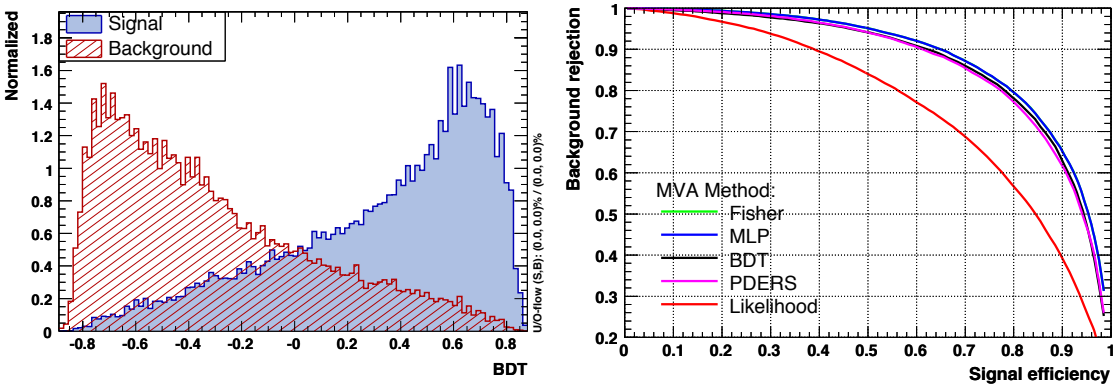

Figure 1: Left: an example of the distributions $g_b(\hat{s}|\nu)$ and $g_s(\hat{s}|\nu)$ when the classifier s is a boosted-decision tree (BDT). Right: the corresponding ROC curve (right) for this and other classifiers. (Figures taken from TMVA manual.)

6.1 Comments on typical usage of machine learning in HEP

Nuisance parameters are an after thought in the typical usage of machine learning in HEP. In fact, most machine learning discussions would only consider $f_b(x)$ and $f_s(x)$. However, as experimentalists we know that we must account for various forms of systematic uncertainty, parametrized by nuisance parameters ν. In practice, we take the classifier as fixed and then propagate uncertainty through the classifier as in Eq. 18. 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 $\nu \neq \nu_0$.

6.2 A more powerful approach

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

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

following the approach outlined in Section 4.

There is an interesting distinction between this approach and the standard use in which the classifier is trained for a fixed 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

$$\frac{f(x|0,\nu_0)}{f(x|\hat{\mu},\nu_0)} = \frac{f_b(x|\nu_0)}{\mu f_s(x_e\,|\,\nu_0) + (1-\mu)\, f_b(x_e\,|\,\nu_0)} = \left[c_1 + c_2 \frac{f_s(x|\nu_0)}{f_b(x_e\,|\,\nu_0)}\right]^{-1} , \tag{20}$$

and c_1 and c_2 are constants. Specifically, the two 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

$$\frac{f(x|0,\hat{\hat{\nu}})}{f(x|\hat{\mu},\hat{\nu})} = \frac{f_b(x|\hat{\hat{\nu}})}{\hat{\mu} f_s(x_e\,|\,\hat{\nu}) + (1-\hat{\mu})\, f_b(x_e\,|\,\hat{\nu})} , \tag{21}$$

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.

6.3 Decomposing tests between mixture models into their components

It is common that the generative model for the low-level features is a mixture model of several components

$$f(x|\theta) = \sum_c w_c(\theta) f_c(x|\theta) . \tag{22}$$

In the case of particle physics, the distributions $f(x|\theta)$ is not a Gaussian Mixture Model, but mixture of complicated distributions associated to relatively few types of particle interactions. Moreover, when searching for a new particle, the null hypothesis would correspond to some of the coefficients $w_c = 0$ while the alternate "signal-plus-background" hypothesis would have $0 < w_{c\in\text{signal}} \ll w_{c\in\text{background}}$. In some cases $w_{c\in\text{signal}}/w_{c\in\text{background}} < 10^{-6}$, which means the alternate hypothesis is a small perturbation to the null hypothesis. This can be a challenge for typical classifiers because they should devote their capacity to the region where $f_{c\in\text{signal}}(x)/f_{c\in\text{background}}(x)$ is relatively large. Lastly, even when the distributions $f_c(x|\theta)$ are well known, it is often the case that the coefficients are uncertain or treated as completely unknown. These all present challenges to machine learning algorithms that aim to learn $s(x;\theta_0,\theta_1)$.

However, it is possible to re-write the target likelihood ratio between two mixture models in terms of pairwise classification problems.

$$\begin{aligned}
\frac{f(x|\theta_0)}{f(x|\theta_1)} &= \frac{\sum_c w_c(\theta_0) f_c(x|\theta_0)}{\sum_c w_c(\theta_1) f_c(x|\theta_1)} && (23)\\
&= \sum_c \left[\sum_{c'} \frac{w_{c'}(\theta_1)}{w_c(\theta_0)} \frac{f_{c'}(x|\theta_1)}{f_c(x|\theta_0)}\right]^{-1} && (24)\\
&= \sum_c \left[\sum_{c'} \frac{w_{c'}(\theta_1)}{w_c(\theta_0)} \frac{g_{c'}(s_{c,c',\theta_0,\theta_1}|\theta_1)}{g_c(s_{c,c',\theta_0,\theta_1}|\theta_0)}\right]^{-1} && (25)
\end{aligned}$$

The second line is a trivial, but useful decomposition into pair-wise classification between $f_{c'}(x|\theta_1)$ and $f_c(x|\theta_0)$. The third line uses the previous results to relate the high-dimensional likelihood ratio into an equivalent calibrated likelihood ratio based on the univariate density of the corresponding classifier, denoted $s_{c,c',\theta_0,\theta_1}$. In the situation where the only free parameters of the mixture model are the coefficients w_c, then the distributions $g_c(s_{c,c',\theta_0,\theta_1}|\theta)$ are independent of θ and can be pre-computed (after training the discriminative classifier, but before performing the generalized likelihood ratio test).

6.4 Related work

In Clayton Scott; Xin Tong (2013), the authors 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. In (Tommi Jaakkola), the authors 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. In (Bianca Zadrozny), the authors 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.

Mention also (Rajat Raina and McCallum, 2003)

7. 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 feedback to the draft, and Yuri Shirman for reassuring cross checks of the Theorem. 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.

References

Michael I. Jordan Andrew Y. Ng. On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.9829&rank=1.

Charles Elkan Bianca Zadrozny. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.3039.

Robert Nowak Clayton Scott. A Neyman-Pearson approach to statistical learning. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.103.6850&rank=5.

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. doi: 10.1140/epjc/s10052-011-1554-0. URL http://arxiv.org/abs/1007.1727.

Michael I. Jordan Stuart Russell Eric P. Xing, Andrew Y. Ng. Distance Metric Learning, With Application To Clustering With Side-Information. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.6509&rank=4.

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

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.

Andrew Y. Ng Rajat Raina, Yirong Shen and Andrew McCallum. Classification with hybrid generative/discriminative models. 2003. URL http://works.bepress.com/andrew_mccallum/38.

Lawrence K. Saul Sam T. Roweis. Nonlinear dimensionality reduction by locally linear embedding. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.111.3313.

Alan Willsky Sujay Sanghavi, Vincent Tan. Learning graphical models for hypothesis testing and classification. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.368.4311.

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. doi: 10.1016/j.physletb.2012.08.020.

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. doi: 10.1016/j.physletb.2012.08.021.

David Haussler Tommi Jaakkola. Exploiting Generative Models in Discriminative Classifiers. URL http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.7709.

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

Xin Tong. A Plug-in Approach to Neyman-Pearson Classification. *Journal of Machine Learning Research*, 14:3011–3040, 2013. URL http://jmlr.org/papers/v14/tong13a.html.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. pages 609–616, June 2001. URL http://dl.acm.org/citation.cfm?id=645530.655658.