Experiments using machine learning to approximate likelihood ratios for mixture models

K Cranmer[1], J Pavez[2], G Louppe[1] and W K Brooks[3]

[1] Physics Department, New York University, New York, NY 10003, U.S.A.
[2] Informatics Department, Universidad Técnica Federico Santa María, 1680 Av. España, Valparaíso, Chile
[3] Physics Department, Universidad Técnica Federico Santa María and Center for Science and Technology of Valparaíso, 1680 Av. España, Valparaíso, Chile

E-mail: juan.pavezs@alumnos.usm.cl

Abstract. Likelihood ratio tests are a key tool in many fields of science. In order to evaluate the likelihood ratio the likelihood function is needed. However, it is common in fields such as High Energy Physics to have complex simulations that describe the distribution while not having a description of the likelihood that can be directly evaluated. In this setting it is impossible or computationally expensive to evaluate the likelihood. It is, however, possible to construct an equivalent version of the likelihood ratio that can be evaluated by using discriminative classifiers. We show how this can be used to approximate the likelihood ratio when the underlying distribution is a weighted sum of probability distributions (e.g. signal plus background model). We demonstrate how the results can be considerably improved by decomposing the ratio and use a set of classifiers in a pairwise manner on the components of the mixture model and how this can be used to estimate the unknown coefficients of the model, such as the signal contribution.

1. Introduction

In High Energy Physics (HEP) and many other fields, hypothesis testing is a key tool when reporting results from an experiment. Likelihood ratio tests are the main technique for hypothesis testing and they are the most powerful statistic for simple hypothesis testing. For composite hypothesis testing the profile or generalized likelihood ratio test is commonly used. When computing the likelihood ratio the data distribution $p(x|\theta)$ must be evaluated, where θ are parameters of the probability distribution. However, it is common in HEP to have physics simulations that allow to sample high dimensional vectors from the distribution $p(x|\theta)$ while not having a description that can be directly evaluated. Commonly it is impossible or computationally expensive to compute the likelihood ratio in this setting.

A common use of likelihood ratios in HEP is signal process identification. In this task the hypothesis testing procedure is used to evaluate the signal process significance by contrasting the background-only (null) hypothesis versus the signal plus background (alternative) hypothesis. In this setting, the underlying distribution can be seen as a signal and background mixture model defined as

$$p(x \,|\, \mu, \nu) = \mu p_s(x \,|\, \nu) + (1 - \mu)\, p_b(x \,|\, \nu), \tag{1}$$

where $p_x(x \mid \nu)$ correspond to the signal distribution and $p_b(x \mid \nu)$ is the background distribution, both parametrized by nuisance parameters ν which describe uncertainties in the underlying physics predictions or response of measurement devices. The parameter μ is the mixture coefficient corresponding to the signal component of the distribution. In this case the generalized likelihood ratio test takes the form of

$$\Lambda(D) = \prod_{e=1}^{n} \frac{p(x_e|\mu = 0, \hat{\hat{\nu}})}{p(x_e|\hat{\mu}, \hat{\nu})}, \tag{2}$$

where D is a data set of i.i.d observations x_e, $\hat{\hat{\nu}}$ is the conditional maximum likelihood estimator for ν under the null hypothesis θ_0 ($\mu = 0$) and $\hat{\nu}, \hat{\mu}$ are the maximum likelihood estimators for ν and μ. This approach has been used extensively to assert the discovery of new particles in HEP [1], such as in the discovery of the Higgs boson [2, 3].

As previously mentioned, the original distributions for signal and background can only be approximated by simulations. Most of the likelihood ratio tests at the LHC are made on the distribution of a single feature that discriminates between signal and background observations. For this, the simulated data is used together with interpolations algorithms in order to approximate the parametrized model and then use it in the hypothesis testing procedure [4].

Recently, it has been shown that a discriminative classifier trained to classify between signal and background can be used to obtain an equivalent likelihood ratio test (see eq. 2.9 of [5]). Given a classifier trained to learn a monotonic function of the per event ratio $p(x_e|\theta_0)/p(x_e|\theta_1)$, it can be proved that the likelihood ratio test on the conditional distributions of the classifier score is equivalent to the original likelihood ratio. Moreover, many of the commonly used classifiers learn to approximate some monotonic function of the per-event ratio.

In this work we show how these results can be used to approximate the likelihood ratio when the underlying distribution is a weighted sum of probability distributions (mixture model). We also show that by training a set of classifiers in a pairwise manner on the components of the mixture model it is possible to improve the results of the approximation.

2. Decomposed likelihood ratio test for mixture models

A generalized version of the signal and background mixture model of eq. (1) for several components is

$$p(x|\theta) = \sum_{i=1}^{k} w_i(\theta) p_i(x|\theta), \tag{3}$$

where $w_i(\theta)$ are the mixture coefficients for each one of the components parametrized by θ. In [5] it is shown that the likelihood ratio between two mixture models

$$\frac{p(x|\theta_0)}{p(x|\theta_1)} = \frac{\sum_{i=1}^{k} w_i(\theta_0) p_i(x|\theta_0)}{\sum_{j=1}^{k'} w_j(\theta_1) p_j(x|\theta_1)}, \tag{4}$$

is equal to the composition of pairwise ratios for each one of the components which, in turn, is equivalent to the composition of ratios on the score distribution of pairwise trained classifiers

$$\frac{p(x|\theta_0)}{p(x|\theta_1)} = \sum_{i=1}^{k} \left[\sum_{j=1}^{k'} \frac{w_j(\theta_1)}{w_i(\theta_0)} \frac{p_j(x|\theta_1)}{p_i(x|\theta_0)} \right]^{-1} = \sum_{i=1}^{k} \left[\sum_{j=1}^{k'} \frac{w_j(\theta_1)}{w_i(\theta_0)} \frac{p_j(s_{i,j}(x;\theta_0,\theta_1)|\theta_1)}{p_i(s_{i,j}(x;\theta_0,\theta_1)|\theta_0)} \right]^{-1}. \tag{5}$$

In the case that the only free parameters of the mixture model are the coefficients $w_i(\theta)$, then each distribution $p_i(s_{i,j}(x;\theta_0,\theta_1)|\theta)$ is independent of θ and can be pre-computed and used afterwards

in the evaluation of the likelihood ratio. Moreover, when numerator and denominator correspond to the same distribution, the values can be directed replaced by unity, avoiding unnecessary computations. Also, for a two-class classifier the values of $s_{j,i}(x;\theta_0,\theta_1)$ for one of the classes can be replaced by the values of $s_{i,j}(x;\theta_0,\theta_1)$ for the opposing class, in which case it is only necessary to train the classifiers for $i < j$. This saves a lot of computation time and reduces the variance that is introduced by differences between $s_{i,j}(x;\theta_0,\theta_1)$ and $s_{j,i}(x;\theta_0,\theta_1)$ due to imperfect training. In the case of background-only versus signal plus background hypotheses it is common that the signal coefficient $w_j(\theta_1)$ is a very small number compared to the background coefficients. In these conditions a classifier trained on data from the full mixture model will have difficulty identifying the signal since most of the useful discriminative data will be located in a small region of the feature space while the decomposed model will not face these issues.

It is possible to estimate the signal and background coefficients by using the maximum likelihood method on the ratios, as follows

$$\hat{\theta} = \arg\max_{\theta} \prod_{e=1}^{N} \frac{p(x_e|\theta)}{p(x_e|\theta_1)} = \arg\max_{\theta} \prod_{e=1}^{N} \frac{p(s(x_e;\theta,\theta_1)|\theta)}{p(s(x_e;\theta,\theta_1)|\theta_1)}, \tag{6}$$

for some fixed value of θ_1.

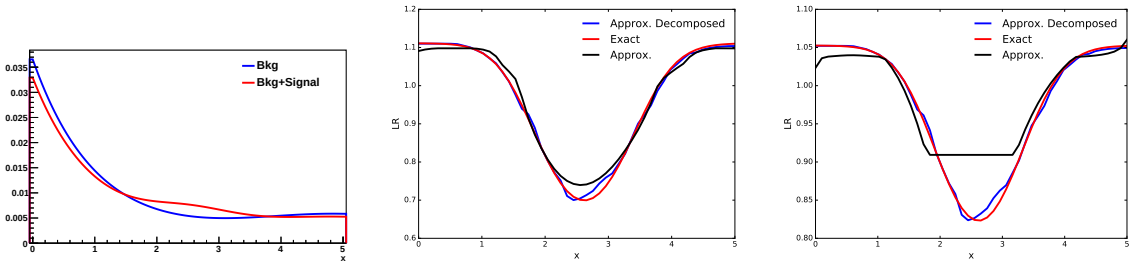
The complete algorithm to approximate the likelihood ratio using pairwise trained classifiers can be separated into three independent stages: classifier training, score distribution estimation, and computation of the composition formula. Since these steps are independent, each one can be solved as a different problem. In the first step any classifier satisfying the monotonic requirement can be used. In the second stage the probability distribution of the score on data from θ_0 or θ_1 can be approximated using any univariate density estimation technique such as histograms or kernel density estimation [6, 7].

3. Experiments

In this section two examples of how the method works on data generated from known distributions will be presented. In both cases we study the results of using the decomposition formula to compute the ratios. Then, we compare the results to the exact (in this case known) ratios and to the density ratios obtained by training a classifier on data from the full mixture model. All studies were conducted using a simple multilayer perceptron model. That classifier shows a good tradeoff between quality of the ratios and simplicity of the model (good results were also obtained using boosted decision trees, logistic regression and support vector machines). The probability models were implemented with the `RooFit` [6] probabilistic programming language and the classifiers were implemented using `Theano` [9] (a framework to build neural network models) and `scikit-learn` [10] (a general framework for machine learning in python). The code is available for replication of the results at [11].

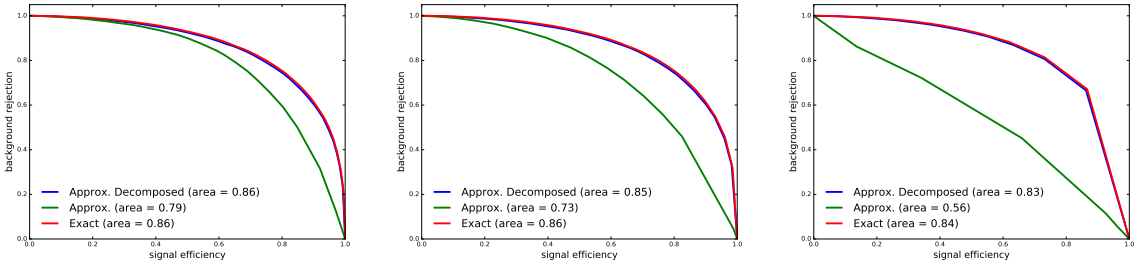
First, we present a simple case in which each component is a univariate distribution. We consider a mixture model consisting of three distributions, where $p_0(x)$ and p_1(x) are univariate Gaussian distributions while $p_2(x)$ is a decaying exponential. The mixture models are composed by the weighted sum of those distributions where $p_0(x)$ correspond to the signal component. The mixture models with coefficients $W(\theta_0) = [0., 0.3, 0.7]$ for the background-only hypothesis and $W(\theta_1) = [0.1, 0.27, 0.63]$ for the background plus signal hypothesis are shown in Figure 1a.

Three neural networks were trained on 200000 examples sampled from the pairs of single distributions and one on examples sampled from the full models (it is important to use the same amount of training data in order to allow a fair comparison). Each neural network consists of 1 hidden layer of 4 units and stochastic gradient descent with a learning rate of 0.01 and no regularization was used in training (the networks were implemented with `Theano`). The distribution of the score is estimated using histograms on a different dataset with 100000



(a) (b) Signal weight of 0.1. (c) Signal weight of 0.05.

Figure 1: (1a) shows the mixture model distributions. (1b) and (1c) show a density ratio comparison for different values of the signal weight.



(a) Signal weight of 0.1. (b) Signal weight of 0.05. (c) Signal weight of 0.01.

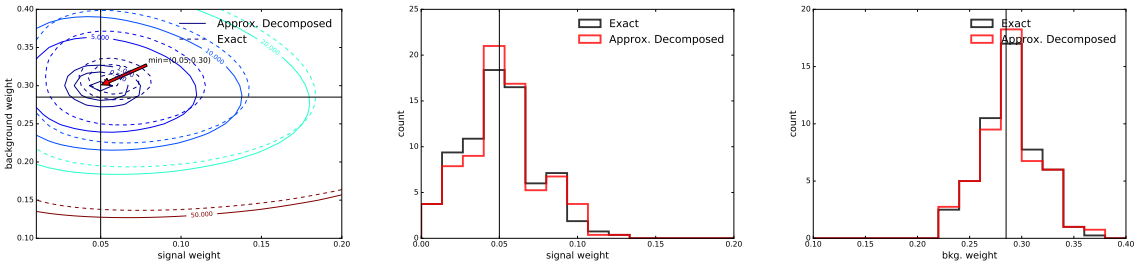
Figure 2: Background rejection versus signal efficiency curve comparison for different values of the signal weight.

samples (the number of bins were carefully chosen in order to allow a good approximation while minimizing Poisson fluctuations). The composed ratio using eq. (5), the ratio estimated using a classifier trained on data from the full model, and the exact density ratio are shown in Figure 1b and Figure 1c for different values of the signal contribution (0.1 and 0.05) while keeping the ratio between the backgrounds contributions fixed.

It can be seen that the ratios obtained by the composition method are better (closer to the exact ratios) than those obtained using a model trained on data from the full mixture models and this become clearer when the signal contribution is smaller.

Next, the same experiments are repeated but now considering a much harder mixture model consisting of three distributions, each one composed of the sum of three 10-dimensional multivariate Gaussian distributions. Again, we used neural networks with 1 hidden layer with 40 units, a learning rate of 0.01 and a ℓ_2 regularization of 0.0001 in training. To evaluate the approximated ratios the background rejection versus signal efficiency curves are used, employing the density ratio as discriminative variable. The values for each one of the three cases (decomposed, full and exact) are shown in Figure 2, for a signal contribution of 0.1, 0.05 and 0.01.

The values of each one of the coefficients of the mixture model can be estimated by using the method of maximum likelihood as explained in Section 2. In Figure 3a the contour plot for the log-likelihood ratio obtained using the composed density ratios and the exact density ratios are shown. Histograms of the maximum likelihood estimators (MLEs) for approximated and exact ratios and for the signal coefficient and one of the background coefficients on 200 different pseudo-datasets of size 1000 are shown in Figure 3b and Figure 3c. It is seen from these histograms that the estimations are unbiased.



(a) $-2\log\Lambda(D)$ (b) Signal weight MLEs. (c) Bkg. weight MLEs.

Figure 3: (3a) shows the values of $-2\log\Lambda(D)$ given the signal and one of the bkg. weights. (3b) and (3c) show histograms for maximum likelihood estimated values for the signal and background weights respectively.

4. Conclusions

We have shown the power of using discriminative classifiers in order to approximate the likelihood ratio. In the case of mixture models we have proved that the decomposed version of the ratio can greatly improve the quality of the results. Using the same method we demonstrated how to estimate the unknown parameters of the model. Initial experiments have been conducted on simulated data from different Higgs production mechanisms, obtaining encouraging results. An open source `Python` package was created to facilitate the usage of this method [12].

5. Acknowledgments

KC and GL are supported by the DIANA grant ACI-14503. JP and WB were partially supported by the Center for Science and Technology of Valparaíso (CCTVal) under Fondecyt grant BASAL FB0821. JP would like to thank Hector Allende and Carlos Valle for helpful discussions.

References

[1] Cowan G, Cranmer K, Gross E and Vitells O 2010 *Eur.Phys.J.* **C71** 1554 (*Preprint* `1007.1727`) URL `http://arxiv.org/abs/1007.1727`

[2] The ATLAS Collaboration 2012 *Phys.Lett.* **B716** 1–29 (*Preprint* `1207.7214`)

[3] The CMS Collaboration 2012 *Phys.Lett.* **B716** 30–61 (*Preprint* `1207.7235`)

[4] Cranmer K, Lewis G, Moneta L, Shibata A and Verkerke W 2012 CERN-OPEN-2012-016, `http://inspirehep.net/record/1236448`

[5] Cranmer K, Pavez J and Louppe G 2016 *ArXiv e-prints* `http://arxiv.org/abs/1506.02169` (*Preprint* `1506.02169`)

[6] Verkerke W and Kirkby D P 2003 *eConf* **C0303241** MOLT007 (*Preprint* `physics/0306116`)

[7] Cranmer K S 2001 *Comput. Phys. Commun.* **136** 198–207 (*Preprint* `hep-ex/0011057`)

[8] Verkerke W and Kirkby D 2003 *Statistical Problems in Particle Physics, Astrophysics and Cosmology, edited by L. Lyons and MK Ünel* 186–190

[9] Bergstra J, Breuleux O, Bastien F, Lamblin P, Pascanu R, Desjardins G, Turian J, Warde-Farley D and Bengio Y 2010 Theano: a CPU and GPU math expression compiler *Proceedings of the Python for Scientific Computing Conference (SciPy)* oral Presentation

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

[11] Pavez J 2016 DecomposingTests: Reproducible results for ACAT 2016 `http://dx.doi.org/10.5281/zenodo.47945`, `https://github.com/jgpavez/DecomposingTests/tree/ACAT2016`

[12] Louppe G, Cranmer K and Pavez J 2016 carl: a likelihood-free inference toolbox `http://dx.doi.org/10.5281/zenodo.47798`, `https://github.com/diana-hep/carl`