Class Project 1

Joel Fisher & Patrick Ley
Machine Learning Course

***Abstract*—The first project revolves around analyzing a data set used to find the Higgs boson. With Stocastic Logistic Regression a model is created that is trained to be a reliable prediction of the measured event.**

I. Introduction

In 1964 Peter Higgs and six other scientists proposed the existence of a particle that explains why particles have mass. In 2013 this theory was confirmed at CERN using the Large Hadron Collider by smashing protons into one another[1]. Even though the Higgs boson decays faster than is observable, scientist were able to measure its decay products.

Using estimation methods learned during the lecture we aim to create a model that accurately predicts which "decay signatures" corresponded to decomposition of a Higgs boson. By using training data that corresponds to a list of features describing the decay signature of a collision event, the model is optimized to maximize the probability of a correct prediction. Since this entails a binary classification, Regularized Logistic Regression is used to train the model.

II. Models and Methods

The prediction of our model should give us a binary response based on the initial collision vectors, thus meaning we will use a binary classification. The training data lets us choose our model family, in our case this being the sigma function $\sigma()$ with polynomial feature expansion of degree d and regularization parameter λ to correct over-fitting.

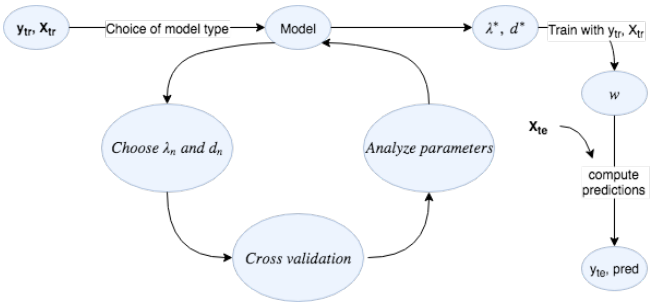

Figure 1. Signal compression and denoising using the Fourier basis.

In a first step we take certain hyper-parameters ($[\lambda, d, ...] \in (\lambda, d, ...)$) to describe our model. The initial parameters are not necessarily the ones that are best suited for the task, so using cross validation their values are analyzed, thus allowing for the tweaking of the $[\lambda, d, ...]$. By splitting the training data into K different parts we can use one of the K parts for training and the rest for testing. Doing this for all by changing the part that trains at every step and by averaging the results, cross validation gives a unbiased estimate of the generalized error and its variance. This process is repeated a couple of times to guarantee a choice of parameters that is close to optimal with respect to the training data.

The hyper-parameters are then used to create a model comprising of the weights w and X_{tr}, y_{tr}. This allows us to calculate the predictions of the given test data.

III. Results

The training methods that gave us the best results was the stochastic version of logistic regression, it is the version that is executed when executing run.py. When testing with least squares and normal logistic regression we got models that were similar in performance, but the least squares, especially the SGD (stochastic gradient descent) version of least squares, were much faster to calculate, yet they didn't perform as well as the aforementioned training method.

A. Choice of hyper-parameters

To find the optimal regularization parameter λ and a good choice for the degree of the polynomial expansion d, cross validation was performed using the two most promising training methods, least squares using normal equations and the stochastic version of the logistic regression. To save computation time the training data set was reduced to roughly 10'000 samples which then was again split into four training subsets. For every tested value of λ and d the weights of the model are compute once with each of the training sets, using the remaining three to compute an estimate of the general loss.

An additional (binary) hyper-parameter of model family was whether or not the feature expansion included mixing of features. Tests with the two selected training methods showed, that this does improve the accuracy of the model. However the actual benefit is relatively small and, since feature mixing of the second order results in an additional 465 features, it comes at a relatively high price in terms of computation speed.

While the least squares method using normal equations gave the best results for comparatively small training sets and, it proved to be impractical when computing the weights

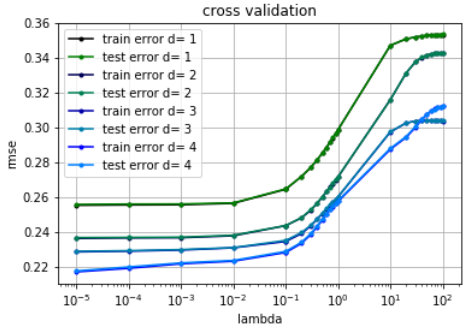

Figure 2. Error of the model using polynomial feature expansion up the degree d and regularization parameter λ trained using normal equations. In the considered range the error diminishes with raising degree and the best fits are generated with small λ.

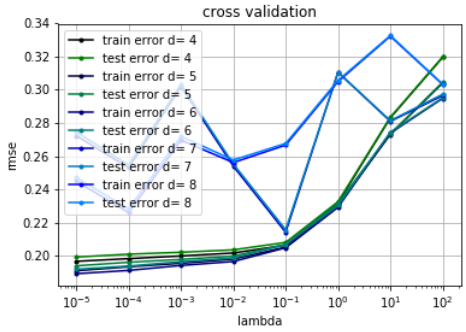

Figure 3. Error of the model using polynomial feature expansion up the degree d with second order degree mixing trained using normal equations. One notices that up to the $d = 6$ the error generally diminishes. Above this value the method becomes unstable.

from the entire training data where a stochastic approach is more suitable. In consequence the cross-validation procedure described above was repeated, as shown in figure 4.

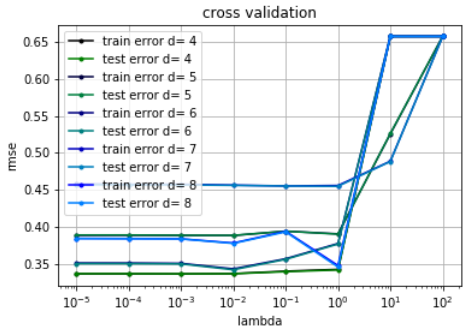

Figure 4. Error of the model using polynomial feature expansion up the degree d with second order degree mixing trained with a stochastic version of logistic regression (batch size = 10). The smallest error can be achieved with $d = 4$.

IV. Discussion

As we can see when looking at the graphs, the regularization parameter λ was kept at zero for most of the experiments as it had no to a negative impact on our calculated training error. We assume this is due to the fact that the data has a small amount of features when compared to the amount of data itself, therefore λ is not needed to correct resulting errors due to over fitting.

The degree influences our error in a predictable manner. The polynomial expansion creates a larger amount of features therefore when it increases so too does the risk of over fitting and with it the error. We can see that when raising the degree above a certain value the error starts to vary quite significantly, especially after degree seven.

75% accuracy leaves room for improvement of our model and the methods used to calculate them. When looking at our

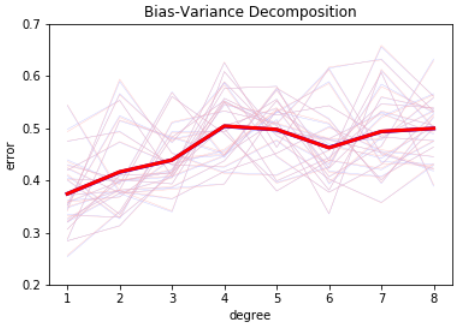

Figure 5. Error for different randomly sampled training sets.

bias-variance diagram we can see that it does not follow the form that we would have expected. For one the test error and the training error seem to be exactly the same when we would have expected the training error to be smaller as the training data error was minimized. On top of that when increasing the degree the training error should decrease as the model more and more accurately describes the training data.

V. Summary

When using the methods learned in course we managed to create a model that gives a passable prediction of the actual detection of a Higgs boson. Cross validation helped us choose the best hyper-parameters for our training methods allowing us to improve the prediction.

References

[1] T. C. Collaboration, "Observation of a new boson at a mass of 125 gev with the cms experiment at the lhc," *Phys. Lett. B*, vol. 716, no. 30, 2012.