Detecting Overfitting Using Unlabelled Data

Tony Liu

1 Context

Overfitting – which we will define generally as the over-representation of model performance – can occur in many ways: training on too few examples, using too complex of a model, leakage of examples between training and validation sets, human-in-the-loop hyperparameter selection, etc. In all of these cases, the model is fit to the noise and structure unique to the data sample, so it will not generalize to unseen data. To combat overfitting, the standard procedure for algorithm evaluation is to train on one dataset and test performance on a distinct held-out set. This assumes there is enough labelled data at the researcher's disposal for both the training and testing sets. However, in some problem domains ground truth labels are expensive to acquire while unlabelled data is plentiful: text is widely available but sentiment analysis often requires manual annotation, unlabelled brain images are cheap to obtain while brain disorder diagnosis requires expensive biopsies, etc. It would be useful to use unlabelled data to detect overfitting as a way to internally validate a model. Though overfit models are characterized as having "too much" variance in bias-variance tradeoff analysis, we see that the distribution of predictions made by overfit models will actually have lower dispersion on a held-out set than on the training set. Intuitively, an algorithm should be more confident on data it has overfit to than on data is has previously not seen. Here we present a statistical test for overfitting based on this idea and evaluate its effectiveness on simulated data.

2 Setup

In a standard classification setting, we assume there is a true underlying data distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the feature space while $\mathcal{Y} = \{0, 1\}$ is the label space. Our goal is to fit a model f which produces a predicted probability that minimizes the loss over the population distribution $\mathcal{D}$, assuming symmetric loss:

$$L_{\mathcal{D}}(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathbf{1}(f(x) > 0.5) \neq y]$$

Since $\mathcal{D}$ is unknown, we rely on measured performance on samples from the distribution to evaluate the model. We have available to us a fully labelled training set $D_{\text{train}} = (X_{\text{train}}, Y_{\text{train}})$ drawn from $\mathcal{D}$ and an unlabelled testing set X_{test} drawn from $\mathcal{X}$. We train on D_{train}, producing a model $\hat{f}$. We can then produce predictions on both the training and testing data by applying $\hat{f}$: $\hat{Y}_{\text{train}}$ and $\hat{Y}_{\text{test}}$. If $\hat{f}$ is overfit, we expect the distribution of test predictions $\hat{Y}_{\text{test}}$ to differ from the distributions of training predictions $\hat{Y}_{\text{train}}$, so detecting overfitting amounts to testing the hypothesis of whether this is the case. We can use the nonparametric two-sample Kolmogorov-Smirnov (KS) test to compare the distribution of predictions, where the null hypothesis is that the two samples come from the same distribution.

3 Preliminary Simulation Results

We evaluate this idea with simulated data and logistic regression with L2 regularization, using the regularization parameter C to control model complexity – lower values of C correspond with stronger regularization. Data are generated using scikit-learn's `make_classification` function, which creates clusters of points normally distributed with variance 1 about the vertices of a d-dimensional hypercube, where d is the number of informative features. We generate a two-class dataset over 100 trials with 500 training and 500 testing samples, $d = 50$ informative features and 450 uninformative features, with $C = [10^{-4}, 10^{4}]$. We see from Figure 1 that for $\alpha = 0.05$ we reject the null hypothesis that the training and testing predictions come from

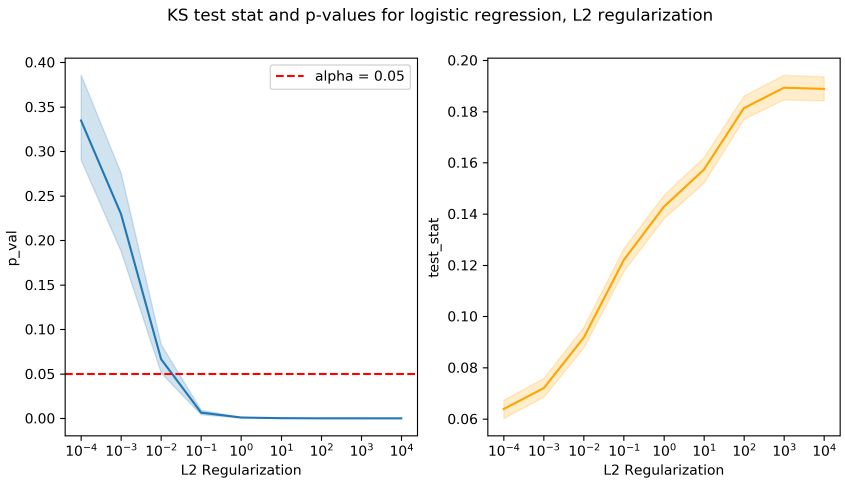

Figure 1: Plots of p-values and test statistic values of the KS test comparing training predictions and testing predictions as a function of regularization strength.

the same distribution at regularization parameter levels $C > 10^{-2}$. This coincides with the optimal setting of C in terms of out-of-sample test performance (Figure 2).

Next steps are to evaluate the statistical power of this test in comparison to the typical method of examining training and testing accuracies.

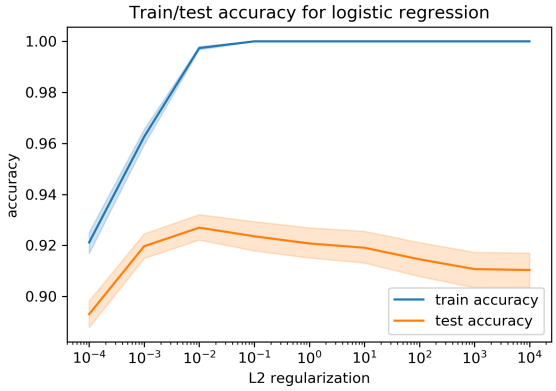

Figure 2: Training and testing accuracy plots as a function of regularization strength.

4 Further Work

Though we have presented this test in terms of internal validation where we have access to both the labelled training set and unlabelled held-out set, we could also use this idea for external validation. For example, if the model and test prediction probabilities were published in addition to test performance metrics as part of a paper submission, we could perform the same prediction distribution comparison to see if submitted models are overfit. Other extensions could involve evaluating how well models generalize to other similar data distributions in the context of covariate drift.