Stop Wasting Time: On Predicting the Success or Failure of Learning for Industrial Applications

J. E. Smith and M. A. Tahir

School of Computer Science
University of the West of England
Bristol, BS161QY, UK
{james.smith,muhammad.tahir}@uwe.ac.uk

Abstract. The successful application of machine learning techniques to industrial problems places various demands on the collaborators. The system designers must possess appropriate analytical skills and technical expertise, and the management of the industrial or commercial partner must be sufficiently convinced of the potential benefits that they are prepared to invest in money and equipment. Vitally, the collaboration also requires a significant investment in time from the end-users in order to provide training data from which the system can (hopefully) learn. This poses a problem if the developed Machine Learning system is not sufficiently accurate, as the users and management may view their input as wasted effort, and lose faith with the process. In this paper we investigate techniques for making early predictions of the error rate achievable after further interactions. In particular we show how decomposing the error in different components can lead to useful predictors of achievable accuracy, but that this is dependent on the choice of an appropriate sampling methodology.

1 Introduction

The successful application of machine learning techniques to industrial problems places various demands on the collaborators. Vitally, the collaboration also requires a significant investment in time and commitment from the end-users in order to provide training data from which the system can (hopefully) learn. This poses a problem if the developed Machine Learning system is not sufficiently accurate, as the users and management may view their input as wasted effort, and lose faith with the process.

A significant factor that would help in gaining confidence and trust from end-users would be the ability to quickly and accurately predict whether the learning process was going to be successful. Perhaps more importantly from a commercial viewpoint, it would be extremely valuable to have an early warning that the user can save their effort while the system designer refines the choice of data, algorithms etc. In a system applied in industrial application, it is necessary that the learning phase is as short as possible and it is essential that the system can tell by itself and as early as possible whether the learning process will be successful. In some cases such as random training input by the operators or missing information the system will not be able to successfully complete the learning process.

In this paper we investigate techniques for making early predictions of the error rate achievable after further interactions. We will consider that we are given N samples, and that the system is still learning and refining its model at this stage. We are interested in predicting what final accuracy might be achievable if the users were to invest the time to create M more samples. In particular we focus on the following aspects:

- What are suitable descriptors of the system's behaviour after some limited number N of samples?
- What is the most appropriate measure of the system's predictive accuracy after further training with $N + M$ samples?
- Is it possible to find useful relationships for predicting the second of these quantities from the first?
- What is the effect of different choices of methodology or sampling regime to estimate these quantities?

In general the error will be a complicated function, but the hypothesis of this paper is that we can deal with it more easily if we decompose it into a number of more stable functions. Therefore this paper concentrates on the use of the well-known bias-variance decomposition as a source of predictors [1, 2]. Although we will use results from a wide range of classifiers, for the purposes of this paper we will take them one at a time, rather than considering heterogenous ensembles of classifiers with different biases. We will also take the pragmatic approach of constructing an "early warning system". In other words, rather than trying to predict the absolute value of the final accuracy, we will consider ways of estimating upper bounds on the accuracy achievable.

The rest of this paper proceeds as follows. In Section 2 we review related work in the field, in particular the bias-variance decomposition of error that we will use. Following that in Section 3 we discuss various suggested methods for using the available data to estimate the various quantities involved. Section 4 describes our experimental methodology and Section 5 describes and discusses the results obtained. Finally in Section 6 we draw some conclusions and suggestions for further work.

2 Bias-Variance Decomposition: A Review

A number of recent studies have shown that the decomposition of a classifier's error into bias and variance terms can provide considerable insight into the prediction of the performance of the classifier [1, 2]. Originally, it was proposed for regression [3] but later, this decomposition has been successfully adapted for classification [1, 2, 4]. While a single definition of bias and variance is adopted for regression, there is considerable debate about how the definition can be extended to classification [1, 5–9]. In this paper, we use Kohavi and Wolpert's [1] definition of bias and variance on the basis that it is the most widely used definition [10, 11], and has strictly non-negative variance terms.

2.1 Basic Definitions of Bias, Variance and Noise:

Kohavi and Wolpert define bias, variance and noise as follows [1]:

Squared Bias: "This quantity measures how closely the learning algorithm's average guess (over all possible training sets of the given training set size) matches the target".

Variance: "This quantity measures how much the learning algorithm's guess bounces around for the different training sets of the given size".

Intrinsic noise: "This quantity is a lower bound on the expected cost of any learning algorithm. It is the expected cost of the Bayes-optimal classifier".

2.2 Kohavi and Wolpert's definition of Bias and Variance

For a particular target function f and a size of the training set m, the expected misclassification rate E(C)(an error has cost 1 and a correct prediction cost 0) is defined as

$$E(C) = \sum_{x} P(x)(\sigma_x^2 + bias_x^2 + variance_x) \tag{1}$$

where

$$\begin{aligned} bias_x^2 &= \frac{1}{2}(1 - \sum_{y \epsilon Y}[P(Y_F = y|x) - P(Y_H = y|x)]^2) \\ variance_x &= \frac{1}{2}(1 - \sum_{y \epsilon Y} P(Y_H = y|x)^2) \\ \sigma_x^2 &= \frac{1}{2}(1 - \sum_{y \epsilon Y} P(Y_F = y|x)^2) \end{aligned}$$

Here x ranges over the instance space X, Y is the predicted variable with elements $y \epsilon \{0, 1\}$ [12]. The actual target function f is a conditional probability distribution and the hypothesis or model h generated by learner is also conditional probability distribution $P(Y_H = y|x)$. Although not clear from the equations, the conditional events in the conditional probabilities are parameterised over f and m. In other words, $P(Y_H = y|x)$ must be rewritten as

$$P(Y_H = y|f, m, x) = \sum_{d} P(d|f, m)P(Y_H = y|d, x) \tag{2}$$

where $P(d|f, m)$ is the probability of generating training sets d from the target f, and $P(Y_H = y|d, x)$ is the probability that the learning algorithm predicts y for point x in response to training set d.

2.3 Bias as an upper limit on accuracy

An alternative perspective on the above analysis is that the bias term reflects an inherent limit on a classifier's accuracy resulting from the way in which it forms decisions boundaries. For example even in a two-dimensional space, an elliptical class boundary can never be exactly replicated by a classifier which divides the space using axis-parallel decisions. Therefore we can treat the sum of the inherent noise and bias terms as an upper limit on the achievable accuracy for a given classifier. A number of studies have been made confirming the intuitive idea that the size of variance term drops as the number of training samples increases, whereas the estimated bias remains more stable, e.g. [2]. Please note that in many prior works it is assumed that the inherent noise term is zero, and also for a single classifier it is not possible to distinguish between inherent noise and bias, so we will adopt the convention of referring to these collectively as bias.

The hypothesis of this paper is that if we can estimate the value of the bias term it will form an accurate predictor to bound the error rate observed after more training examples. The way that we will do this for a given size sample N is to repeatedly draw test and training sets from the sample and observe what proportion of the items are always misclassified, what proportion are sometimes misclassified, and what proportion are never misclassified. As we will next discuss, this raises the issues of how we should do this repeated process.

3 Prediction Methodology

As discussed in Section 2, a number of recent studies have shown that the decomposition of a classifier's error into bias and variance terms can provide considerable insight into the prediction of the performance of the classifiers [1, 2]. However, identifying the quantities that we wish to measure merely leads us to the next question - what is the most appropriate methodology for estimating the values of those quantities?

To give a simple example of why this is important, the hypothesis of this paper relies on being able to distinguish between those data items that are **always** going to be misclassified by a given classifier, and those which will **sometimes** be misclassified, depending on the choice of training set. Since the well known $N-$fold cross-validation approach only classifies each data item once, it does not permit this type of decomposition and cannot be used. Luckily alternative approaches have been identified and studies by other authors. Leveraging this work, in this paper, we will compare the approaches proposed by Kohavi & Wolpert [1] and Webb & Conilione [10].

Kohavi & Wolpert Hold-out Procedure: Kohavi & Wolpert used a holdout procedure for estimating the bias and variance of a classifier C from a dataset D. In their approach, samples D are randomly divided into 2 parts: Training samples T_r and Testing samples T_e. T_r samples are further divided into N training sets $t_{r_1}, t_{r_2}, ..., t_{r_n}$ by uniform random sampling without replacement. To get training set of size m, they chose T_r to be size $2m$. That allows $(\frac{2m}{m})$ different possible training sets; and thus guarantees that there are not many duplicate training sets in the set of N training sets; even for small values of m. Each classifier is trained using each training sets and bias and variance are estimated using test set T_e. The outcome of this is a set of N class precisions for each of the elements in the test set.

Webb & Conilione sub-samples Cross Validation Procedure: In the second set of experiments, we will decompose error into bias-variance using sub-sampled cross-validation proposed by Webb & Conilione [10] but using same definitions for bias and variance as above. Webb & Conilione have argued that hold out approach proposed in [1] is fundamentally flawed and resulting in small training sets and thus provide instability in the estimates it derives. They proposed that sub-sampled cross-validation (CV) procedure is superior to both the holdout and bootstrap procedures and thus provides greater degree of variability between training sets. Webb's procedure repeats N-Fold CV l times. This ensures that each sample x of the dataset D is classified l times. The $bias_x$ and $variance_x$ can be estimated from the resulting set of classifications. The final bias and variance is estimated from the average of all $x \epsilon D$ [10, 11].

4 Experimental methodology

Choice of Classifiers: Ten different classification algorithms are selected each with different bias and variance characteristics namely: Naive Bayes [13], Decision Tree [15], Nearest Neighbor [16], Bagging [18], AdaBoost [19], Random Forest [20], Decision Table [21], Bayes Network [13], Support Vector Learning [22], and Ripple-Down Rule learner [17]. All these classifiers are implemented in WEKA library [17].

Data sets: The experiments are carried out on the following Four Artificial and Five Real-World Surface Inspection data sets described in Table 1. Each artificial dataset consists of 13000 contrast images with a resolution of $128 * 128$ pixels. The good/bad labels were assigned to the images by using different sets of rules of increasing complexity. The proposed prediction analysis is also evaluated out on real world data sets of CD-print and Egg inspection. The data set for CD print consists of 1534 images and each image is labeled by 4 different operators. Thus, 4 different CD print data sets are available. From each set of images, we derive 2 feature vectors (FVs) consisting of 17 and 74 features respectively. The first FV contains only image-level information while second FV also contains features from objects within the image.

Trend Line using Linear Regression Linear regression is a statistical tool used to predict future values from past values. Regression lines can be used as

Table 1. Datasets Description

Name	Samples	Description
Artificial 1-3	13000	Used for Linear Regression Analysis
Artificial 4	13000	Used after Linear Regression Analysis for Prediction
CD Print Op1-Op3	1534	Used for Linear Regression Analysis
CD Print Op4	1534	Used after Linear Regression Analysis for Prediction
Egg	4238	Used for Linear Regression Analysis

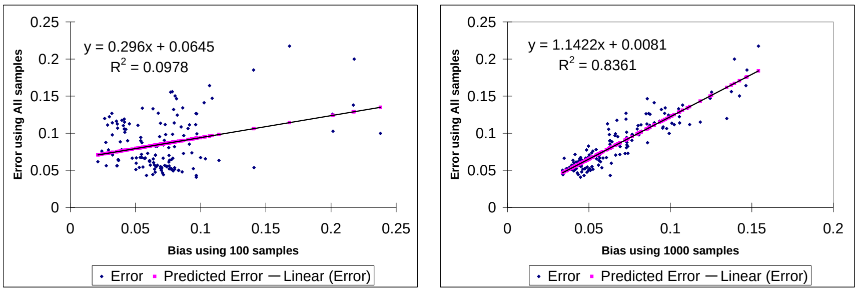
a way of visually depicting the relationship between the independent (x) and dependent (y) variables in the graph. A straight line depicts a linear trend in the data. In this paper, we will use linear trend line between bias (For first N samples only) and error (For $N + M$ samples) to predict the Success or Failure of Learning for Industrial Applications. We will use squared Pearson correlation coefficient R^2 as a measure to analyze the quality of prediction. The closer R^2 is to 1.0; the better is the prediction. This is of course an extremely simple way of measuring the relationship between estimated bias and final error, and more sophisticated techniques exist in the fields of statistics and also Machine Learning. However, as the results will show it is sufficient for our purposes. An obvious candidate for future work is to consider approaches which will give us confidence intervals on the predicted error for a given observed bias, as this will fit in better with the concept of providing an upper bound on the achievable accuracy.

5 Results and Discussion

As discussed in Section 1, we have estimated the bias using $\{100, 200, 300, ...1000\}$ samples and then the error using all samples of artificial/real data sets by both Kohavi and Webb sampling approaches.

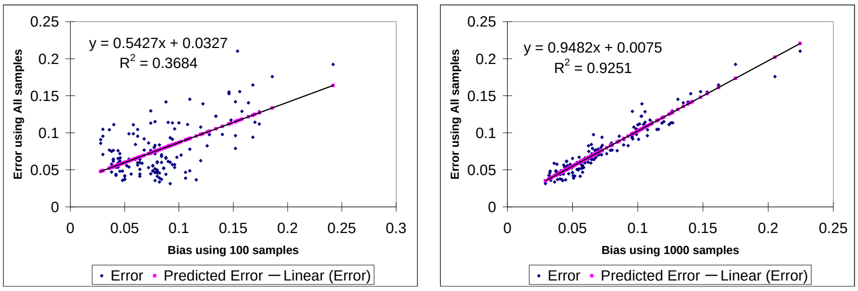
Results with Kohavi's sampling procedure: Figure 1 shows linear regression analysis for bias-error when Kohavi's approach is used for bias-error decomposition. Bias is estimated using 100 and 1000 samples respectively. 7 data sets are used for regression analysis (3 Artificial data sets, CD Print labeled by 3 operators, and 1 Egg data set). Each data set consists of 2 different feature vectors and is evaluated using 10 classes as discussed in Section 4. The goodness of fit of regression model is measured using Correlation R^2. As clearly indicated from these graphs, R^2 is very low when model is fit using only 100 samples while correlation is high when 1000 samples are used. Furthermore, straight line using 1000 samples depicts a linear trend in the data.

Results with Webb's sampling procedure: Figure 2 shows linear regression analysis for Bias-Error when Webb's approach is used for bias-error decomposition. Again, as clearly indicated from these graphs, the R^2 is very low when model is fit using only 100 samples while correlation is high when 1000 samples are used. However, these values are consistently higher than those obtained using Kohavi's approach.



(a) (b)

Fig. 1. Graphs showing linear regression analysis for Bias-Error using 100 and 1000 samples respectively. Kohavi approach is used for bias-error decomposition.

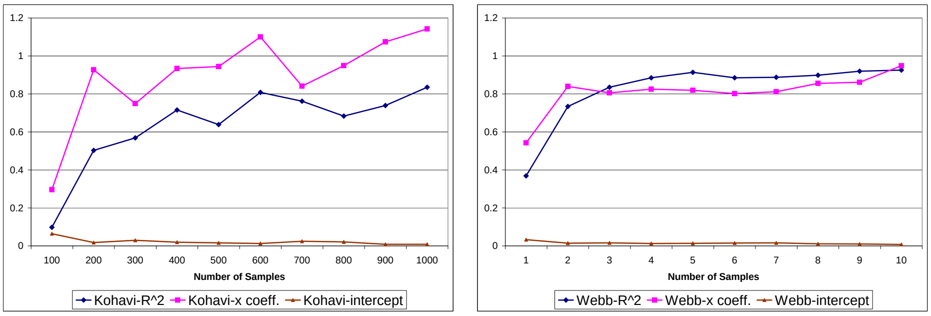


(a) (b)

Fig. 2. Graphs showing linear regression analysis for Bias-Error using 100 and 1000 samples respectively. Webb approach is used for bias-error decomposition.

Stability of Predictions: Figure 3 shows the graph indicating relationship between varying number of samples {100,200,300,...,1000} and {R^2, x-coefficient, intercept} for both Kohavi's and Webb's approaches. This shows how rapidly the linear regression equation stabilizes in these two cases. It is clear from the graph that correlation using Webb approach is high and more stable. One of the reasons that Kohavi's approach is not stable is the use of hold out approach. It has been argued that in Kohavi's approach, samples are randomly divided into training and testing pools and then training pool is further divided into training sets and that can results in instability in the estimates [11]. Another explanation is that a single test set is chosen from the available samples. For small sample sizes this may not always contain sufficiently representative set of items so successive test sets might be "easy" or "hard".

Prediction Testing using Trained/Unseen Datasets from Trained Regression Model: In our experiments, R^2 is used as a measure to evaluate the goodness of regression models. Another way to evaluate the goodness of regression models is as follows:



(a) (b)

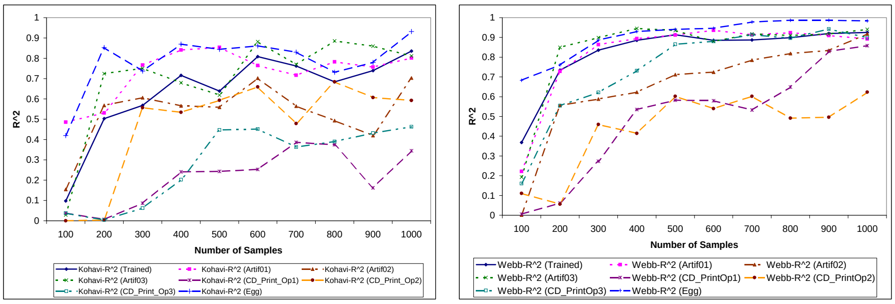
Fig. 3. Number of samples vs $\{R^2$,x-coeff.,intercept$\}$ using (a) Kohavi's (b) Webb's bias-error decomposition.

- For each combination of the ten classifiers, the seven data sets used in training, the two unseen data sets, and the ten sample sizes we repeat the following:
  - estimate the bias component of the error using both Kohavi and Webb's approaches
  - plug this value into the regression equations obtained above to get a predicted final error
  - compare this to the observed final error
- these values can now be subjected to a new regression analysis to see how well the predictions correlate to observed error from the full dataset.

Figures 4 and 5 show the relationship between number of samples and R^2. Again, it is clear from both these figures that R^2 using Webb approach is high and stable. Artif04 has approximately the same R^2 as that of trained one especially when samples are greater than 500 by regression equation using Webb sampling approach. Correlation is quite low for CD-Print-Op2 and CD-Print-Op4 data sets. Closer inspection of the results shows that in almost every case the observed error from 1534 samples of the CD print data is higher than that predicted by inserting the bias observed from fewer samples into the regression equation. This is because the observed error still contains a significant component of variability due to the effects of the relatively small training and test sample sizes. By contrast, for the artificial data sets, where we have nearly ten times more data, the variance components have almost disappeared and so our predictions correlate highly to the observed error. This illustrates our earlier point - that the predictions we are making here have to be treated as upper bounds on the achievable accuracy.

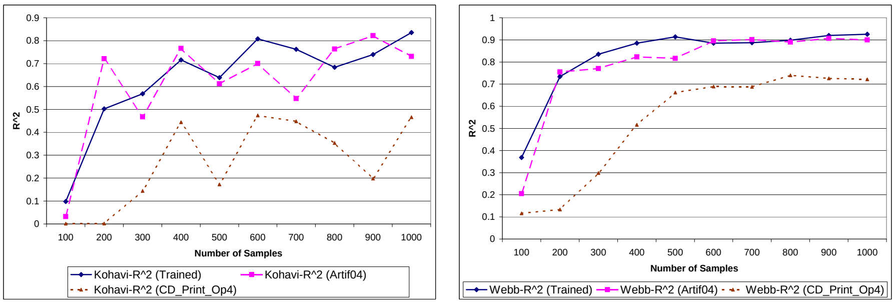
6 Conclusion

In this paper, we have investigated techniques for making early predictions of the error rate achievable after further interactions. Linear trend line between



(a) (b)

Fig. 4. Number of samples vs R^2 for trained data using (a) Kohavi's (b) Webb's approach.



(a) (b)

Fig. 5. Number of samples vs R^2 for unseen data using (a) Kohavi's (b) Webb's approach.

bias and error is used to predict the Success or Failure of Learning for Industrial Applications. The experiments are carried out on the Artificial and Real-World data sets. We have shown that squared Pearson correlation coefficient R^2 is a good measure to analyze the quality of prediction. We have also shown that Webb's approach allows much more accurate and stable estimates of error components. These results are valid for ten very different forms of classifier used in this paper. As the high correlation between the long-term observed error, and the predictions for that based on observed bias after 1000 samples shows, the (bias+implicit noise) term of the error stabilises rather quickly for all of the different types of classifier tested. Thus it forms not only a good upper bound on the achievable accuracy, but also a good estimator for the final accuracy provided enough samples are available for the variance term to decrease.

This is of course an extremely simple way of measuring the relationship between estimated bias and final error, and more sophisticated techniques exist in the fields of statistics and also Machine Learning. However, as the results will show it is sufficient for our purposes. An obvious candidate for future work is to consider approaches which will give us confidence intervals on the predicted error for a given observed bias, as this will fit in better with the concept of providing an upper bound on the achievable accuracy, that can be used as an "early warning" of impeding failure, so that users' confidence can be maintained.

Acknowledgements

This work was supported by the European Commission (project Contract No. STRP016429, acronym DynaVis). This publication reflects only the authors' views.

References

1. R. Kohavi and D. H. Wolpert (1996), "Bias Plus Variance Decomposition for Zero-One Loss Functions", Proceedings of the 13th International Conference on Machine Learning.
2. D. Brian and G. I. Webb (1999), "On the effect of data set size on bias and variance in classification learning", Proceedings of the 4th Australian Knowledge Acquisition Workshop, 117-128.
3. S. Geman and E. Bienenstock and R. Doursat (1995), "Neural Networks and the bias/variance dilemma", Neural Computation 4, 1-48.
4. J. J. Rodriguez and C. J. Alonso and O. J. Prieto (2005), "Bias and Variance of Rotation-based Ensembles", Computational Intelligence and Bioinspired Systems, Lecture Notes in Computer Science, Springer, 3512, 779-786.
5. L. Breiman, "Bias, variance, and arcing classifiers", Technical report 460, Statistics Department, University of California, Berkeley, CA.
6. P. Domingos (2000), "A unified bias-variance decomposition and its application", In Proceedings of the 17th International Conference on Machine Learning", 231-238, Stanford University, USA.
7. J. H. Friedman (2000), "On bias, variance, 0/1-loss, and the curse of dimensionality", Data Mining and Knowledge Discovery, 1(1), 55-77.
8. G. James (2003), "Variance and bias for general loss functions", Machine Learning, 51(2), 115-135.
9. B. E. Kong and T. G. Dietterich (1995), "Error-correcting output coding corrects bias and variance", In Proceedings of the 12th International Conference on Machine Learning, 313-321, San Francisco, Morgan Kaufmann.
10. G. I. Webb (2000), Multiboosting: A technique for combining boosting and wagging. "Machine Learning", 40(2), 159–196.
11. G. I. Webb and P. Conilione (2003), "Estimating bias and variance from data", http://www.csse.monash.edu.au/ webb/Files/WebbConilione03.pdf, (Under Review).
12. P. I. D. Putten and M. V. Someren (2004) , "A Bias-Variance Analysis of a Real World Learning Problem: The CoIL Challenge 2000", Machine Learning, 57, 177-195.
13. R O. Duda and P. E. Hart and D. G. Stork (2000), "Pattern Classification", 2nd Edition, Wiley Interscience.
14. J. R. Quinlan (1993). "C4.5: Programs for Machine Learning.", Morgan Kaufmann Publishers Inc.
15. A. K. Jain and R. P. W. Duin and and J. Mao (2000). "Statistical Pattern Recognition: A Review.", IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1), 4–37.
16. T. M. Cover, and P. E. Hart (1967). "Nearest Neighbor Pattern Classification", IEEE Transactions on Information Theory, 13(1), 21–27.
17. I. H. Witten and E. Frank (2005). "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco.
18. L. Breiman (1996). "Bagging predictors", Machine Learning, 24(2), 123–140.
19. Y. Freund and R. E. Schapire (1996). "Experiments with a new boosting algorithm, Proceedings of International Conference on Machine Learning, 148–156.
20. L. Breiman (2001). "Random Forests, Machine Learning, 45(1), 5–32.
21. R. Kohavi (1995). "The Power of Decision Tables. Proceedings of the 8th European Conference on Machine Learning.
22. J. Platt (1998). "Fast Training of Support Vector Machines using Sequential Minimal Optimization". Advances in Kernel Methods - Support Vector Learning, B. Schoelkopf, C. Burges, and A. Smola, eds., MIT Press.