Practical Bias Variance Decomposition

Remco R. Bouckaert

Computer Science Department, University of Waikato, New Zealand
rrb@xm.co.nz, remco@cs.waikato.ac.nz

Abstract. Bias variance decomposition for classifiers is a useful tool in understanding classifier behavior. Unfortunately, the literature does not provide consistent guidelines on how to apply a bias variance decomposition. This paper examines the various parameters and variants of empirical bias variance decompositions through an extensive simulation study. Based on this study, we recommend to use ten fold cross validation as sampling method and take 100 samples within each fold with a test set size of at least 2000. Only if the learning algorithm is stable, fewer samples, a smaller test set size or lower number of folds may be justified.

1 Introduction

Is the improved performance of C4.5 with increasing training set sizes due to decrease in bias, decrease in variance, or both? Figure 1 shows how bias variance decomposition changes with increasing training set size. The first three plates are three separate runs of decompositions according to Kohavi [1] (using Weka [2] with default settings, i.e. 50 samples). Everything was kept the same except that a different randomization of the data set was used. The first plate suggests both variance and bias decreases with increasing training set size. The second plate suggests bias decreases but variance remains the same. The third plate suggests bias remains unchanged while variance decreases. This raises the following questions: why are these decompositions so different and how to select the correct decomposition.

The reason for the conclusions of these decompositions to be so radically different is because the decomposition tends to be sensitive to the particular randomization of the data set. Clearly, this is not desirable. Furthermore, the example shown in Figure 1 is not a hand picked example or fluke of the data. The plates shown are selected from among the first ten runs and the dataset was generated from a naive Bayes data source (see Section 4). It did not take long to find such contradicting outcomes.

Clearly, it is important for a bias variance decomposition to show low variability of the estimates, since it impacts conclusions drawn from them. The randomization of the dataset is not the only issue that impacts the variability of the decomposition. In the literature (Section 2.1), various methods for bias variance decomposition are proposed, but no two papers seem to agree on the parameters of an empirical method. In this paper, we investigate these issues and perform an empirical investigation in order to be able to recommend a sound way to perform bias variance decompositions that minimizes variability of the bias and variance estimates. Based on our findings, the fourth plate in Figure 1 turns out to be the correct interpretation.

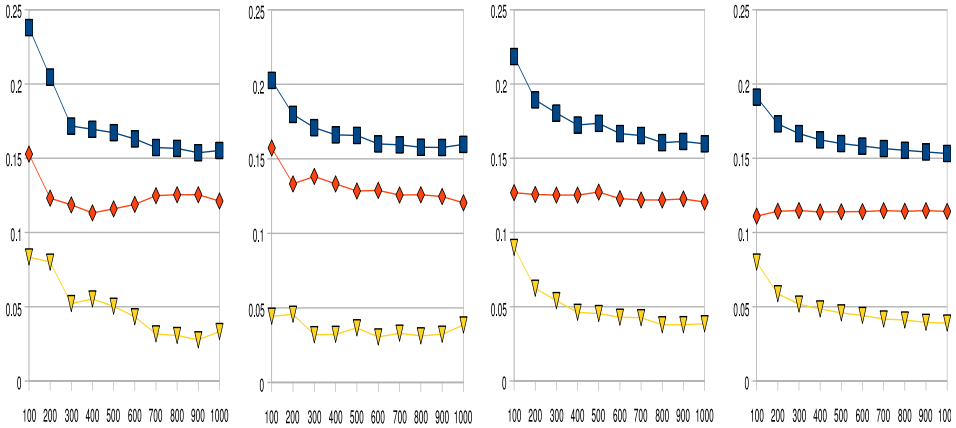

Fig. 1. Various bias variance decompositions on the same problem (see text for description). Training set size on x-axis, error (top line) bias (middle line) and variance (bottom line) on y-axis.

2 Bias Variance Decomposition

Bias variance decomposition described in the literature follows the following pattern (illustrated in Figure 2). A *data source* is a provider of *training sets*. Data sources can be synthetic where a model of the domain is used to create new independent training sets every time a new set is requested. In practice, we are more interested in data sources that take samples of fixed data sets. The training sets obtained from the data source are used to train a learning algorithm. The thus leaned models are applied on a *test set*. The bias and variance are then estimated for each instance in the test set and for 0-1 loss the bias and variance are calculated from the number of incorrect learners for the instance. The bias for an instance is estimated as $\sum_i (x_i - p_i)^2 - p_i * (1 - p_i)/(n - 1)$ where i sums over the class values, n is the number of learners applied to the instance, x_i an indicator variable (0 or 1) that is indicates whether the instance class value equals the ith value, and p_i the fraction of learners that correctly predicted x_i. The variance for an instance is estimated as $1 - \sum_i p_i^2$. The final bias and variance reported are averaged over the instances in the test set.

2.1 Experimental methodology from the literature

In the literature, the following methods for bias variance decomposition can be found.

Kong and Dietterich [3] use a synthetic data source for 200 data sets of 200 instances and a test set of 7670 instances were drawn for direct estimates of bias and variance.

In Kohavi and Wolpert [1], a sample of size n without replacement from a data set D is taken to get a training set source. From the remainder, a test set of size n

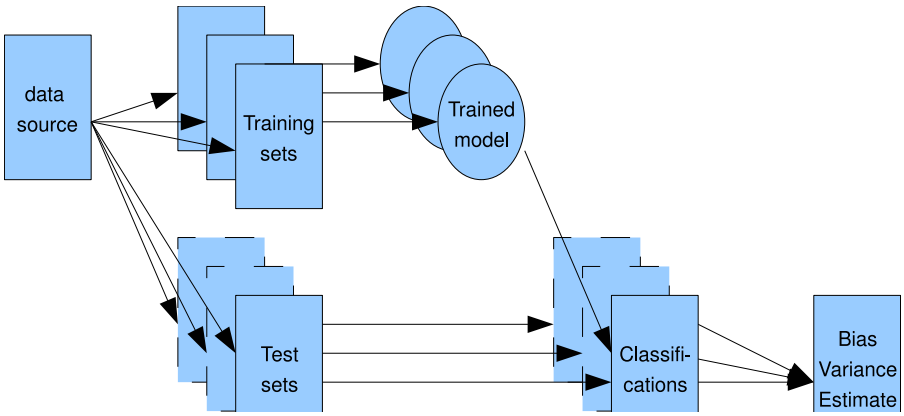

Fig. 2. General overview of bias variance decomposition estimation.

is sampled without replacement. The training source set is used to draw a sample of size $m < n$ on which the learning algorithm is trained. There are 50 such training samples drawn. The resulting 50 trained models are then applied to the test set and the bias and variance of the learner are calculated from the predictions on the test set.

Domingos [4] splits data sets randomly into training and test sets with a proportion of 2:1. From the training set 100 bootstrap samples (i.e. samples with replacement) were taken and the learner trained on the 100 samples and applied to the test set.

Bauer and Kohavi [5] use the technique of [4] but repeated three times and the estimates are averaged over the three attempts.

James [6] uses a synthetic model to generate 100 and 1000 data sets of a (unknown) fixed size and a large (unknown sized) test set in order to obtain bias and variance estimates. For UCI data sets, 5-fold cross-validation was used and from the non-test set 50 bootstrap samples were drawn. Bias and variance were averaged over the folds. One UCI data set comes with a separate test set, and for this one no cross validation was used but just a single run was done.

Valentini and Dietterich [7] use synthetic data sources to generate 100 training sets of 200 and 400 instances and a test set of 10.000 instances. Also, real data was split into data source and test set where 200 data sets of 100 instances are drawn with replacement from the data source.

Valentini and Dietterich [8] select 200 bootstrap samples (with replacement) from a data set, trains algorithms on the data set and applies the classifier to samples not selected for the training set (the so called 'out of bag' set). For each instance, a bias and variance is estimated and the final bias and variance estimates are obtained by averaging over the instances.

Webb [9] uses 10 times repeated 3 fold cross validation and averages results over runs and folds. From each train set source, 100 samples are drawn for training sets.

Webb and Conilione [10] use repeated (10 and 50 times) fold cross validation where the training set source is sampled from the non-test set. Parameters for training set size and training set dependency can be provided for the training sets that are sampled from the training source.

3 Problems identified

It appears that every researcher uses her own favorite method for doing bias variance measurement experiments, which makes comparison among various works hard. Also, it appears that ad-hoc methods are used when the experiments do not behave well. In particular when estimates turn out to be insufficiently stable the number of data sets is arbitrary increased till the estimates appear stable.

The following concrete problems can be identified;

Gold standard Synthetic data sources can be created so that beforehand the systematic error (noise, Bayesian rate) can be determined. However, it is not clear how the type of data source impacts on the bias and variance of variance. For example, the LED data generator [11] which can be interpreted as a naive Bayes model for generating data sets may result in more stable estimates than when a decision tree is used as data source. Also, the number of attributes, amount of noise in the attributes, cardinality of nominal attributes, presence of numerical attributes and other data source characteristics may influence experimental results.

Empirical standard No generally accepted protocol exists for measuring bias and variance for a given data set. The main features in which methods differ are the following;

- Variants for obtaining data sources are train/test split of data, cross validation and bootstrap.
- The way data source are used to create data sets using sampling with or without replacement. Sampling with replacement results in data sets with possibly different characteristics than the original data set. Also, it impacts on some learning algorithms, for instance, 1-nearest neighbor.
  Another way data source use varies is that some instances can be ignored in order to control the interdependency between sampled data sets.
- The number of data sets sampled from each data source, which ranges from 50 to 1000 in the literature.
- The number of times the process is repeated to generate more stable estimates, e.g. once or 10 times repeated cross validation.

Desiderata Most articles do not explicitly list the properties of a bias variance estimator that are desirable. In this article, we concentrate on the following criteria;

- *Unbiased estimates.* Just as for any estimator.
- *Stability/replicability.* Repeated measurements on the same data with different random splits of this data should result in (almost) equal estimates.

– *Efficiency.* As little computational effort should be required for estimating bias and variance.
– *Data source control.* Most bias variance measurements are applied to a particular 'real world' data set of size n but no method allows for sampling data sets of size n (without duplicating data). The size of the data sets is one item to control. Another is dependency between data sets.

4 Simulation Study

To examine the parameters mentioned in the previous section, we performed a number of experiments. We considered a range of data sources.

Bayesian network data sources For data sources, Bayesian networks with increasing numbers of complexity were randomly generated. Figure 3 shows the network structures and marginal distributions. The first has the same topology as Naive Bayes, the second is a tree augmented naive Bayes (TAN) structure and the third is a general Bayesian network structure, each structure representing increasingly complex concepts. The networks have 10 binary variables each and 50%/50% class probabilities.

Agrawal data sources Agrawal et al. [12] defined a set of ten functions for machine learning benchmarking. These are functions over the attributes salary, commission, age, education level, car, zipcode, house value, years house owned and total loan amount. The functions are classification functions splitting the population into two groups and can be as simple as splitting on age in the interval 40 to 60 years. Others are more complex functions like testing whether $0.67(salary + commission) - 5000 \cdot eduction\ level + 0.2 \cdot equity - 20.000$ is positive where *equity* is a hidden variable calculated as 0 if the house is owned less than 20 years or $0.1 \times house\ value \times (years\ house\ owned - 20)$ otherwise.

Gaussian radial base data sources By creating a random set of centers with randomly assigned weights, a set of random Gaussian radial base functions can be defined around those centers. By selecting a center at random according to the weights of the centers and generating attributes randomly offset from the center and assigning the class attribute associated with the center, new instances can be generated. This type of data generator is called the random RBF generator. In the experiments, random RBF data generator from Weka [2] were used with 10, 20, 50 and 100 centers.

Number of samples To eliminate dependence between samples on bias/variance estimates, initially we used the data sources to produce new independent samples every time a sample is required. After determining other parameters, sampling methods (resampling, cross validation and bootstrapping) based on single data sets are considered.

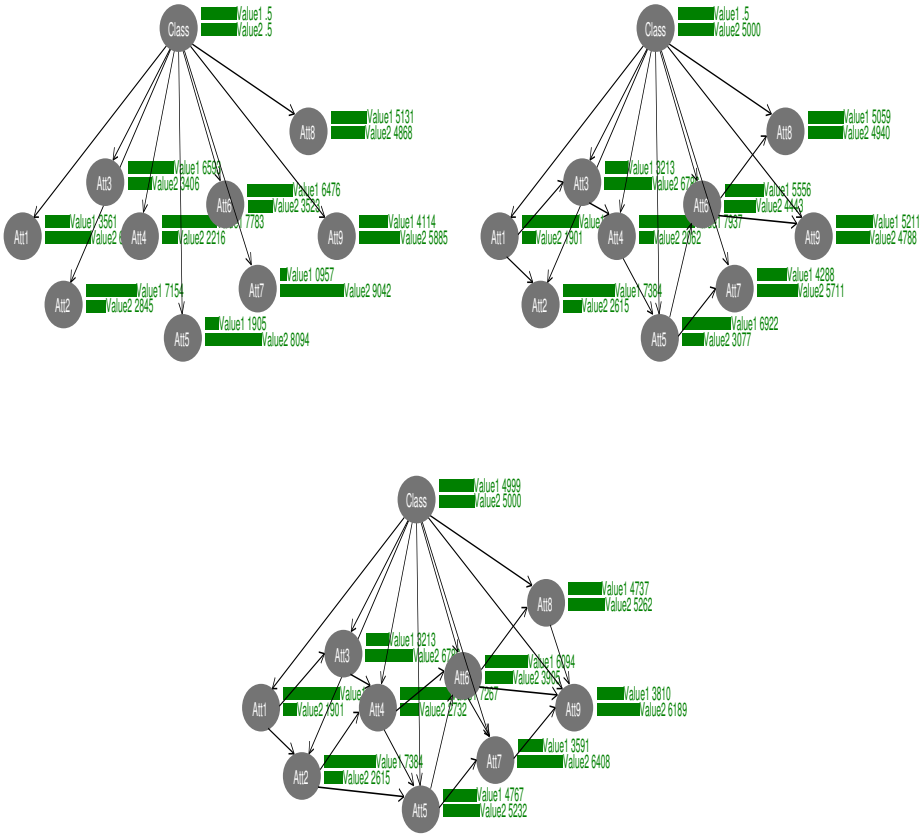

Fig. 3. Bayesian networks used as data generators.

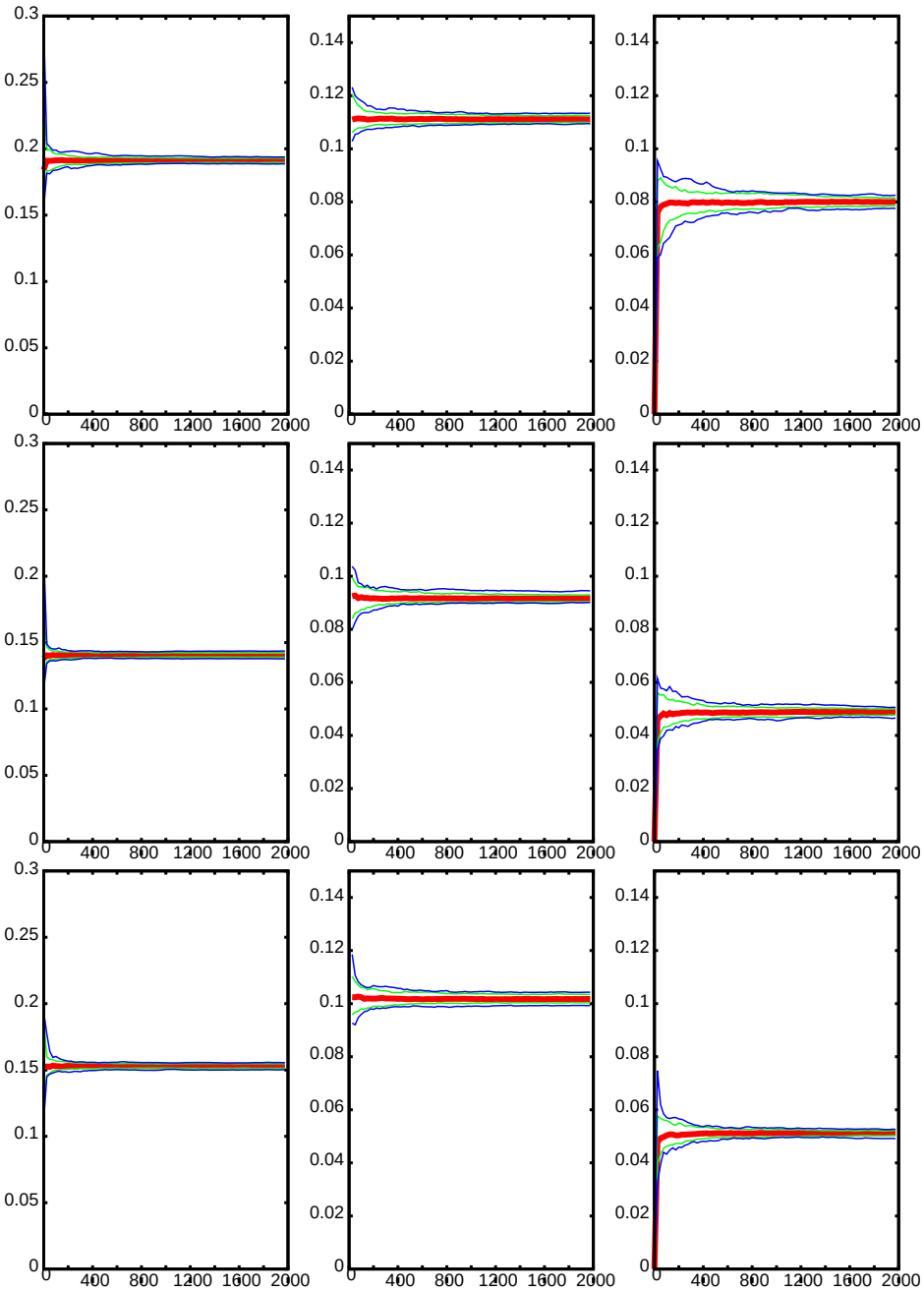

Fig. 4. Comparing different complexity of concepts on bias/variance for C4.5; naive Bayes data source in top row, TAN in middle row and full Bayesian net in bottom row. Error (left), bias (middle) and variance (right) estimates and their 100% and 90% bounds on the y-axis. Number of datasets sampled on the x-axis (ranging 0 to 2000).

The first experiment aims at determining the number of samples required for getting a stable estimate of the bias and variance. Also, it tries to get an impression on the impact of the data source on the variability of bias/variance estimates. Up to 2000 samples were drawn from the Bayesian networks and C45 as implemented in Weka [2] was trained and then tested on a set of 10.000 instances. This process was repeated 100 times for the three networks, so 60.000 trees were trained in this experiment. Figure 4 shows the results.

The same experiment as for the naive Bayes, TAN and Bayesian net data generator was repeated for the ten Agrawal functions (for which a total of 10x100x2000= 200.000 trees were learned) and with the random RBF data generators (for which a total of 4x100x2000= 80.000 trees were learned). However, results are not shown due to space limitations.

In general, the experiments show that taking less than 200 samples tends to have a significant impact on the variability of the bias/variance estimates. This implies most results published in the literature (as outlined in Section 2.1) can be expected to suffer from unstable estimates. The variability in the estimates does not decrease very much when taking more than 1000 samples. The variability of the estimates for large number of samples differs with every data source. However, as Figure 4 shows, the variability does not necessarily increase with increasingly complex data sources. The above observations hold for the ten Agrawal data sources and random RBF generators as well.

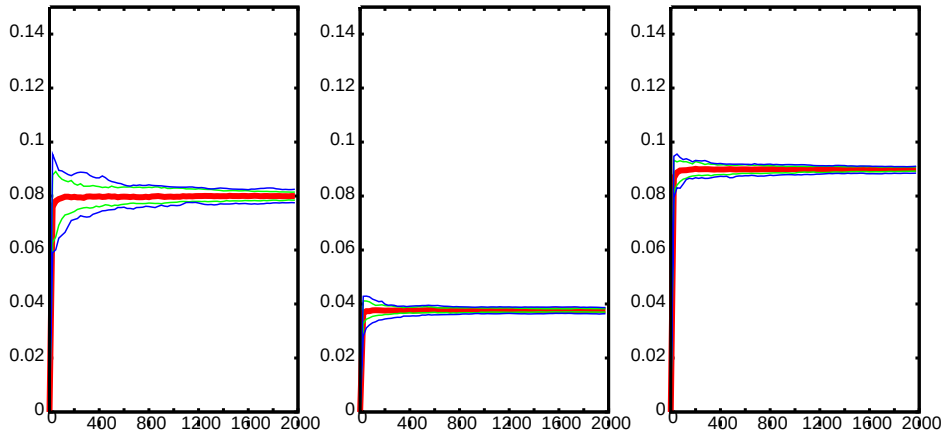

Fig. 5. Comparing various learning algorithms; left C4.5, middle Naive Bayes, right nearest neighbor. Variance estimates and their 100% and 90% bounds on the y-axis. Number of datasets sampled on the x-axis (ranging 0 to 2000).

Learning algorithm To get an impression how sensitive the bias variance decomposition is to the actual learning algorithm under investigation, we reran the above experiment with C4.5, naive Bayes and nearest neighbor as learning algorithms. Figure

5 shows a result typical for the outcomes for the naive Bayes data source (remainder not shown due to limited space). Clearly, different algorithms have different bias and variance, but also the variability of the estimates differs with different algorithms. Stable algorithms like naive Bayes appear to result in less variability than highly unstable ones like C4.5 [13]. Nearest neighbor is a medium stable algorithm and has variability of estimates in between the other two algorithms. The same observations hold for the other data sources.

Test set size The test set size can be expected to have an impact on the bias variance decomposition since small test sets can be expected to result in highly variable decompositions. Likewise, virtual infinite test set sizes can be expected to result in more stable decompositions. To get a sense where the desired balance between stable decompositions and acceptable computational effort lies, we ran the experiment with different test set sizes and Figure 6 shows the results. Clearly, test set sizes under 2000 result in highly variable decompositions. Improvement in the stability of the decomposition vanishes for test sets over 10.000 instances.

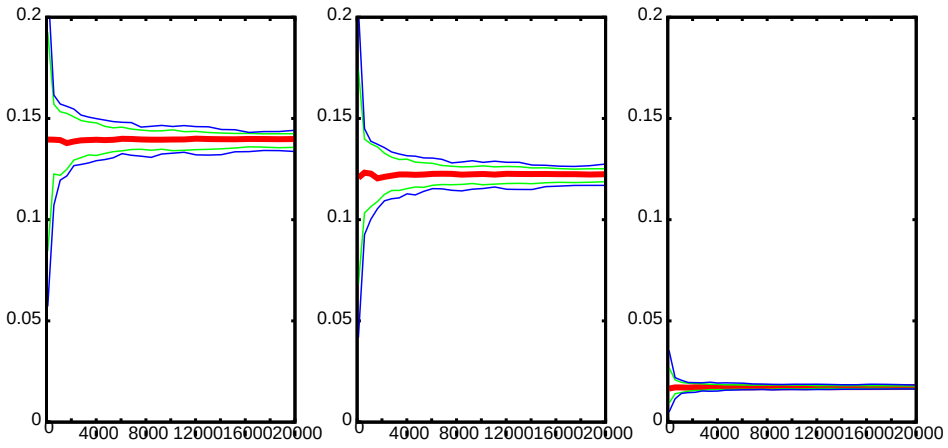

Fig. 6. Effect of test set size on stability of error/bias/variance estimates. On x-axis the test set size from 100 to 20.000 instances. Naive Bayes data source. On y-axis the error (left plate), bias (middle) and variance (right) with median, 90% interval and 100% interval over 100 runs.

Sampling method The sampling method was reported to have some effect on the variability of the bias variance estimates [10], in particular cross validation appeared to result in more stable outcomes than sampling with replacement. Another approach is bootstrapping with out of bag instance classification for estimates [8]. Figure 7 shows the mean and 90% interval for error (left plate), bias (middle) and variance

(right) estimates using sampling with replacement (first pair of items in plate), cross validation (next pair of items) and bootstrapping (last pair). The estimates are for C4.5 on the naive Bayes data source from Section 4 and the training sets contain 1000 instances. Computationally, resampling is cheapest, cross validation takes a bit more due to some extra administration and bootstrapping takes even more due to the administration that comes with out of bag estimation. The first item in each pair is where 1000 samples are taken from the data source, and the second where 10.000 samples were taken. For cross validation, ten folds were used and within each fold 100 samples were taken. To get the 10.000 samples the process was repeated ten times and estimates averaged over the ten runs. Figure 7 shows clearly that the variability for cross validation is considerably less than that for resampling, confirming [10]. Furthermore, it appears that bootstrapping results in higher variability of bias/variance estimates. Furthermore, increasing the number of samples has little effect on the variability of the estimates, so this variability is inherent in the samplings method.

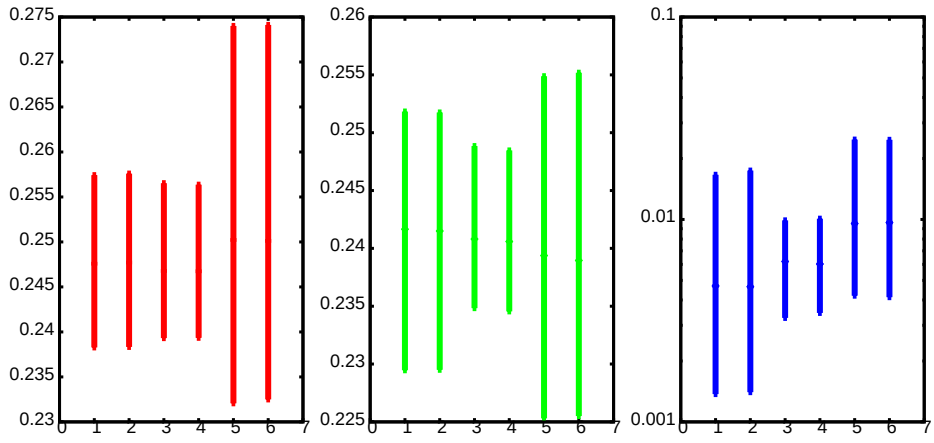

Fig. 7. Mean error (left), bias (middle) and variance (right) with 90% interval for 1000× and 10000× resampling, 1× and 10× 10 fold cross validation and 1000× and 10000× bootstrapping respectively. Note linear y-scale for error and bias and log scale for variance.

5 Discussion/Conclusions

We identified the following issues that have an impact on the stability of bias variance decompositions

– number of samples drawn from a datasource. Decompositions using less than 200 samples result in highly unstable estimates. This is surprising since proposals found in the literature routinely use 100 or less samples. Consequently, it is easy to draw erroneous conclusions from such simulations (as illustrated by Figure 1).

– learning algorithm. Unstable algorithm like C4.5 can result in twice the variability of a decomposition as stable algorithms like naive Bayes. Increasing the number of samples can reduce this effect somewhat, but cannot totally eliminate the variability due to instability of the learning algorithm.
– test set sizes under 2000 result in highly variable decompositions and are not recommended. Test set sizes of 10.000 and over do not seem to reduce variability any more. So, as a rule, the larger the test set size the lower the variability.
– sampling algorithm. Cross validation gives the least variable estimates, bootstrapping the most and resampling goes in between.
– The data source has a small effect on the variability, but no pattern could be found to determine when it can be justified to use fewer samples.

Based on these observations, we recommend to use ten fold cross validation as sampling method and take 100 samples within each fold with a test set size of at least 2000. Taking fewer samples or using a lower number of folds such that the number of samples (i.e. the number of folds times number of samples per fold) is at least 200 may be justified if the learning algorithm under consideration is very stable.

Acknowledgements

I thank Geoff Holmes for the helpful comments on improving the paper.

References

1. Kohavi, R., Wolpert, D.H.: Bias plus variance decomposition for zero-one loss functions. In Saitta, L., ed.: Machine Learning: Proceedings of the Thirteenth International Conference, Morgan Kaufmann (1996) 275–283
2. Witten, I., Frank, E.: Data mining: Practical machine learning tools and techniques with Java implementations. Morgan Kaufmann, San Francisco (2000)
3. Kong, E.B., Dietterich, T.G.: Error-correcting output coding corrects bias and variance. In: Proceedings of the 12th International Conference on Machine Learning, Morgan Kaufmann (1995) 313–321
4. Domingos, P.: A unified bias-variance decomposition and its applications. In: International Conference on Machine Learning, ICML-2000. (2000) 231–238
5. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning **36**(1-2) (1999) 105–139
6. James, G.: Variance and bias for general loss functions. Machine Learning **51** (2003) 115–135
7. Valentini, G., Dietterich, T.G.: Bias-variance analysis and ensembles of svm. In: Multiple Classifier Systems: Third International Workshop. (2002) 222–231
8. Valentini, G., Dietterich, T.G.: Low bias bagged support vector machines. In: International Conference on Machine Learning, ICML-2003. (2003)
9. Webb, G.I.: Multiboosting: A technique for combining boosting and wagging. Machine Learning **40**(2) (2000) 159–196
10. Webb, G.I., Conilione, P.: Estimating bias and variance from data (unpublished manuscript available from `http://www.csse.monash.edu.au/~webb/files/webbconilione06.pdf`) (2002)

11. Olshen, L.B.J.F.R., Stone, C. In: Classification and Regression Trees. Wadsworth International Group, Belmont, California (1984) 43–49
12. Agrawal, R., Imielinski, T., Swami, A.: Database mining: A performance perspective. IEEE Transactions on Knowledge and Data Engineering **5**(6) (1993) 914–925 Special issue on Learning and Discovery in Knowledge-Based Databases.
13. Dwyer, K., Holte, R.: Decision tree instability and active learning. In Kok, J.N., Koronacki, J., de Mántaras, R.L., Matwin, S., Mladenic, D., Skowron, A., eds.: ECML. Volume 4701 of Lecture Notes in Computer Science., Springer (2007) 128–139