



Bootstrap Bias Corrected Cross Validation Applied to Super Learning

Krzysztof Mnich¹✉ , Agnieszka Kitlas Golińska² , Aneta Polewko-Klim² ,
and Witold R. Rudnicki^{1,2,3} 

¹ Computational Centre, University of Białystok,
ul. Konstantego Ciołkowskiego 1M, 15-245 Białystok, Poland
k.mnich@uwb.edu.pl

² Institute of Informatics, University of Białystok,
ul. Konstantego Ciołkowskiego 1M, 15-245 Białystok, Poland

³ Interdisciplinary Centre for Mathematical and Computational Modelling,
University of Warsaw, ul. Tyniecka 15/17, 02-630 Warsaw, Poland
<https://ii.uwb.edu.pl>, <https://icm.edu.pl>

Abstract. Super learner algorithm can be applied to combine results of multiple base learners to improve quality of predictions. The default method for verification of super learner results is by nested cross validation; however, this technique is very expensive computationally.

It has been proposed by Tsamardinos et al., that nested cross validation can be replaced by resampling for tuning hyper-parameters of the learning algorithms. The main contribution of this study is to apply this idea to verification of super learner. We compare the new method with other verification methods, including nested cross validation.

Tests were performed on artificial data sets of diverse size and on seven real, biomedical data sets. The resampling method, called Bootstrap Bias Correction, proved to be a reasonably precise and very cost-efficient alternative for nested cross validation.

Keywords: Super learning · Cross validation · Bootstrap · Resampling

1 Introduction

Numerous machine learning algorithms with roots in various areas of computer science and related fields have been developed for solving different classes of problems. They include linear models, support vector machines, decision trees, ensemble algorithms like boosting or random forests, neural networks etc [1]. There are also many feature selection techniques used to prepare the input data for the predictive algorithm. Different methods can extract and utilise different parts of the information contained in the data set under scrutiny. It has been shown that in many cases an ensemble machine learning model, which combines several different predictions, can outperform the component learners. This can

Algorithm 1: k -fold cross validation $\mathbf{CV}(f, D)$

input : learning method f ,
data set $D = \{y_j, X_j\}, j = 1, \dots, N$

output: N predictions for the response variable $\psi_j, j = 1, \dots, N$

split randomly the data set into k almost equally-sized folds F_i

foreach F_i **do**

- define a training set as $D_i = D \setminus F_i$
- learn a predictive model using the training set $M_i \leftarrow f(D_i)$
- apply the model to the remaining subset $\Psi_i \leftarrow M_i(F_i)$
- collect the predictions Ψ_i

end

be realised in the form of wisdom of crowds [2] or in a more systematic way as a Super Learner [3]. The wisdom of crowds is one of the principles underlying the design of DREAM Challenges, where multiple team contribute diverse algorithms and methodologies for solving complex biomedical problems [4]. Super learning was proposed by van der Laan et al. and implemented as SuperLearner R language package. It utilises cross validation to estimate the performance of the component algorithms and dependencies between them. One may notice that wisdom of crowds can be formally cast as a special example of super learning. The goal of the current study is to examine several methods for estimate the performance of the super learner algorithm. In particular it explores methods for minimising bias with minimal computational effort.

1.1 Super Learning – Basic Notions and Algorithms

The input of any machine learning algorithm is a set of N observations, usually assumed to be independent and identically distributed, of the response variable y_j and a set of p explanatory variables $X_j = \{x_{jm}\}, m = 1, \dots, p$. A predictive algorithm f (that includes also the feature selection algorithm and the set of hyper-parameters) can be trained using a data set $D = \{y_j, X_j\}$ to produce a model M . Then, the model can be applied to another set of variables X' to obtain a vector of predictions $\Psi = M(X')$.

K -fold cross validation is an almost unbiased technique to estimate performance of a machine learning algorithm when applied to unseen data. Algorithm 1 allows to compute a vector of N predictions for all the observations in the data set. The predictions can be compared with the original decision variable Y to estimate the performance of the learning algorithm.

In super learning, we use multiple machine learning algorithms $f_l, l = 1, \dots, L$. The cross validation procedure leads to L vectors of predictions Ψ_l . The idea is to treat them as new explanatory variables and apply some machine learning algorithm to build a second-order predictive model (see Algorithm 2). The new features are expected to be strongly and linearly connected with the response

variable, so linear models with non-negative weights seem to be appropriate super learning algorithms. Indeed they are default methods in SuperLearner R package. Note, however, that any method that selects a weighted subset of the elementary learning algorithms can be formally considered as an example of super learning. This includes also selection of the best-performing algorithm, which is a common application of cross validation, and can be considered as a special case of super learning [3]. Other examples can be unweighted mean of k best performing algorithms, as in the wisdom of crowds, or even a simple mean of all learning algorithms used.

Algorithm 2: Super learning $\mathbf{SL}(f_l, f_c, D, D')$

input : L learning methods f_l ,
 combining method f_c ,
 data set $D = \{Y, X\}$,
 new data set $D' = \{Y', X'\}$

output: ensemble predictive model M ,
 predictions for the new response variable Ψ'

compute L vectors of cross validated predictions $\Psi_l \leftarrow \mathbf{CV}(f_l, D)$

build a second-order data set $D_c \leftarrow \{Y, \Psi_l\}$

apply the method f_c to the second-order data set $M_c \leftarrow f_c(D_c)$

build L predictive models using the entire data set $M_l \leftarrow f_l(D)$

apply the models to the new data $\Psi'_l \leftarrow M_l(D')$

compute the ensemble predictions $\Psi' \leftarrow M_c(\Psi'_l)$

return $M = \{M_l, M_c\}$, Ψ'

1.2 Performance Estimation for Super Learning – Nested Cross Validation

Super learning, as every machine learning method, is sensitive to overfitting. Therefore, an unbiased estimate of the performance of ensemble models is necessary. The obvious and most reliable method to obtain it is an external cross validation. The entire procedure is called nested cross validation, as it contains two levels of CV loop (see Algorithm 3).

Nested cross validation is implemented in SuperLearner R package as a default method of performance estimation. The authors of Super learner algorithm recommend to use 10-fold internal cross validation [3]. The external CV should also be at least 10-fold, to avoid a meaningful reduction of the sample size. Even if one restricts himself to a single loop of the external CV, the entire procedure requires all the learners to be run 110 times. Although the routine is easy to parallelise, the computational complexity is very large.

Algorithm 3: Nested cross validation $\text{NCV}(f_l, f_c, D)$

input : L learning methods f_l ,
 combining method f_c ,
 data set $D = \{Y, X\}$

output: predictions for the response variable Ψ

split randomly the data set into k almost equally-sized folds F_i

foreach F_i **do**

define a training set as $D_i = D \setminus F_i$
 run a super learning procedure $\{M_i, \Psi_i\} \leftarrow \text{SL}(f_l, f_c, D_i, F_i)$
 collect the predictions Ψ_i

end

compare Ψ with corresponding values of Y

1.3 Bootstrap Bias Correction for Super Learning

An alternative approach to verification of complex machine learning algorithms was proposed by Tsamardinos et al. [5]. It is called the Bootstrap Bias Correction and bases on Efron's bootstrapping technique [6]. The method was originally proposed to estimate the bias caused by a choice of the optimal set of hyper-parameters for a predictive model. However, as it has been mentioned, this task can be considered as a special case of super learning. What is more, the procedure is general and does not involve any actions specific to selection of hyper-parameters. Thus, the Tsamardinos' method can be easily generalised for any kind of super learning.

Main Contribution of the Present Study. The purpose of the current study is to develop and test the algorithm of Bootstrap Bias Correction for an arbitrary super learning method. The idea is shown in Algorithm 4. The cross validated predictions for all the algorithms are computed only once. Then, the combining models are computed many times for samples that are drawn with replacement from the predictions. The predictions of each combination model are then tested on it's out-of-bag observations.

In this method, all base predictions come from the same cross-validation run, hence the entire sample is somewhat co-dependent. Therefore learning on the outcome will be overfitted. On the other hand, the training set contains duplicates what introduces additional noise. The effective data set size is smaller than the sample. This effect should decrease the overfitting and cancel the bias of the performance estimate.

The major advantage of BBC method is its small computational complexity. The elementary predictions are computed only once, multiple runs are required only for the relatively simple combining procedure.

Algorithm 4: Bootstrap bias corrected super learning **BBCSL**(f_l, f_c, D)

input : L learning methods f_l ,
 combining method f_c ,
 data set $D = \{Y, X\}$

output: set of pairs of the response variable and predictions $\{y_j, \Psi_j\}$

foreach f_l **do**

 compute the cross validated predictions $\Psi_l \leftarrow \mathbf{CV}(f_l, D)$
 build a second-order data set $D_c \leftarrow \{Y, \Psi_l\}$

end

for $b \leftarrow 1$ **to** B **do**

 draw a random sample D_b of the prediction set D_c with replacement
 define the out-of-bag set $D_{\setminus b} \leftarrow D_c \setminus D_b$
 apply the method f_c to the data set D_b $M_b \leftarrow f_c(D_b)$
 compute predictions for the out-of-bag set $\Psi_b \leftarrow M_b(D_{\setminus b})$
 compare the predictions Ψ_b and with the corresponding values of Y

end

In the current study, we apply bootstrap bias corrected super learning algorithm to synthetic and real-world data sets and compare the results with other verification methods, including the nested cross validation.

2 Materials and Methods

The tests were focused on binary classification tasks. The area under ROC curve (AUC) was used as a quality measure of predictions, since it does not depend on the class balance in the data set. The methods, however, can be applied also for multi-class classification or regression tasks, with different quality metrics.

2.1 Data

Artificial Data. The methodology was developed on the synthetic data set. The data set was created as follows:

- First, the two vectors of expected values and two covariance matrices were randomly generated. These parameter sets are common for all the observations.
- The requested number of instances of the binary decision variable was randomly chosen, with the same probability for both classes.
- For each class of the decision variable, the explanatory variables are drawn from a multivariate normal distribution with the corresponding set of parameters.

This procedure allows to create an arbitrary big sample with the same statistical properties and verify directly the predictions on the unseen data. The parameters were tuned to emulate the strength of linear, quadratic and pairwise interactions as well as the dependencies between variables that may appear in the real-world data sets.

The data sets we had generated contained 5000 explanatory variables and 50, 100, 150, 200 observations. Statistic tests indicate from 2 to 20 relevant variables, depending on the sample size.

Biomedical Data Sets. The tests were performed on seven data sets that contain measurements of gene expression and copy number variation for four cancer types. These data sets correspond to biological questions with different levels of difficulty. These are:

- data sets obtained from the CAMDA 2017 Neuroblastoma Data Integration Challenge (<http://camda.info>):
 - CNV – 39 115 array comparative genomic hybridization (aCGH) copy number variation profiles, data set limited to a cohort of 145 patients,
 - MA – 43 349 GE profiles analysed with Agilent 44 K microarrays, data set limited to a cohort of 145 patients,
 - RNA-seq – 60 778 RNA-seq GE profiles at gene level, data set limited to a cohort of 145 patients.

The data collection procedures and design of experiments were previously described in the original studies [7–11]. Data sets are also accessible in Gene Expression Omnibus [12]. The relevant question for these data sets is predicting the final clinical status of the patient using molecular data. This is difficult problem.

- data sets with The Cancer Genome Atlas database generated by the TCGA Research Network (<https://www.cancer.gov/tcga>) that contain RNA-sequencing data for various types of cancer [13–18]:
 - BRCA RNA – data set contains 1205 samples and 20223 probes (112 normal and 1093 tumor),
 - BRCA CNV – data set contains 1314 samples and 21106 probes (669 normal and 645 tumor),
 - HNSC – data set contains 564 samples and 20235 probes (44 normal and 520 tumor),
 - LUAD – data set contains 574 samples and 20172 probes (59 normal and 515 tumor).

In this case the relevant question is discerning normal tissue from tumor using data set at hand. It is much easier question since both genetic profile and gene expression patterns are highly modified in cancer cells in comparison with normal tissue.

2.2 Methods

For each data set we applied the full protocol of super learning and nested cross validation. We used the default 10-fold setup for both external and internal cross validation and 100 repeats of resampling procedure. For the artificial data sets, we performed the entire routine for 100 different sets drawn from the same distribution. In the case of real data sets, the protocol was repeated 100 times on the same data for different cross validation splits.

Base Machine Learning Algorithms. Six popular machine learning algorithms were used as base learners:

- Random Forest [19],
- LASSO [20],
- Support Vector Machine (SVM) [21],
- AdaBoost [22],
- Naive Bayesian classifier,
- kNN classifier for $k = 10$.

All the algorithms are implemented from the standard R packages, with the default parameters. The parameters used are obviously not optimal, but the performance optimisation is not the subject of the current study.

For each algorithm, the set of input variables was reduced to the most relevant ones by the feature selection algorithm. To this end, we applied Welch t -test for each explanatory variable and chose 100 variables with the biggest value of the test statistic. This procedure is very sensitive to overfitting, so any bias in the verification methods should be clearly visible.

Methods of Super Learning. Six methods of combining various prediction results via super learner approach were tested:

- two default methods implemented in SuperLearner R package:
 - **NNLS:** non-negative least squares
 - **NNlog:** non-negative logistic regression
- two “toy example” methods, that are, however, commonly used:
 - **Mean:** average of all the base results (the method does not introduce any overfitting)
 - **Best 1:** choice of the best-fitted model (this special case is mentioned in the original van der Laan paper and corresponds directly to the original purpose of the bootstrap method by Tsmardinos)
- **Best k :** average of k best-performing models, where k is optimised on the training set – usually k is set as 3–4
- **RF:** Random Forest algorithm

Methods of Verification. The estimate of quality was performed using the following methods (from the most biased up to unbiased one):

- **Training set:** measure the performance of the combined classifier on the same data that were used to build the combined model (the results are obviously overestimated);
- **Independent CV:** compute the results of base learners only once, then verify the combining algorithm in a second, independent cross validation (due to the common information in the training and validation sets the results are also overestimated);
- **BBC SL:** apply the bootstrap bias correction method (the method proposed in the current study),
- **Nested CV:** apply the nested cross validation (as a gold-standard);
- **New data:** directly measure the performance for new data, drawn from the same distribution (the oracle for artificial data sets).

Table 1. The AUC of base classification algorithms for the artificial data. Comparison between prediction estimate in 10-fold cross-validation (**10CV**), prediction of new data for model trained on entire sample (**100%**) and prediction of new data for model trained on 90% of the sample (**90%**) (the same training set size as in cross-validation). The uncertainty values of AUC were computed as mean square error (MSE) of average of 100 independent measurements.

	10CV	100%	90%	10CV	100%	90%
	50 obs., MSE=0.01			100 obs., MSE=0.01		
RandomForest	0.62	0.64	0.62	0.69	0.71	0.69
LASSO	0.62	0.63	0.62	0.68	0.70	0.69
SVM	0.63	0.64	0.62	0.68	0.71	0.69
AdaBoost	0.38	0.33	0.38	0.68	0.71	0.69
Naive Bayes	0.63	0.65	0.63	0.69	0.72	0.70
kNN	0.56	0.57	0.55	0.61	0.63	0.61
	150 obs., MSE=0.005			200 obs., MSE=0.005		
RandomForest	0.735	0.760	0.745	0.782	0.804	0.788
LASSO	0.744	0.772	0.747	0.817	0.853	0.821
SVM	0.738	0.754	0.744	0.777	0.803	0.787
AdaBoost	0.699	0.731	0.711	0.761	0.789	0.763
Naive Bayes	0.735	0.756	0.744	0.770	0.792	0.779
kNN	0.659	0.678	0.672	0.690	0.711	0.698

3 Results

3.1 Artificial Data Sets

Base Learners – Cross Validation vs. Direct Verification The way of building the artificial data sets allows for comparison between the cross validation results and the actual results obtained for unseen data. This comparison is shown in Table 1.

As it could be expected, the performance of all the machine learning algorithms improves with the sample size. In particular AdaBoost could not cope with 50 observations only.

Interestingly, a considerable negative bias of the cross validated estimations of AUC was observed in comparison with the model trained on the entire sample. To correct this difference, additional models were built using the training sets reduced to 90% of the original sample. Quality of these models, agree much better with the cross-validated estimates. That means that most of the bias of the cross validation procedure comes from the smaller size of training sets. The small remaining bias is most likely due to negative correlations of fluctuations in training and validation sets in cross-validation.

Super Learning. The performance of diverse methods of super learning is shown in Table 2. For this particular data set, the super learning technique needs at least 100 observations to outperform the best single result and at least 200 observations to perform better than a simple average of all the base results. Random forest proves to be a poor super learning method. The non-negative linear models perform the best (surprisingly, the ordinary least squares method was better at classification than the logistic regression, that is specialised for this task). However, the simple best- k method performs almost as well.

One should note the difference between the performance of the “Best 1” method in the Table 2 and the best overall performance from the Table 1. The reason is, that the algorithm, that performed best on a subset of the data will be not necessarily the best one for the entire data set. Everyone, who chooses the most appropriate machine learning algorithm using a single cross validation, should thus expect some decreasing of its performance for new data.

The differences between various performance estimates are small, but some regular patterns are noticeable. As previously, the performance of the models learned on the entire data set and applied to new data is better than measured in the nested cross validation. Moreover, the performance measured for models built using the reduced training set is slightly better as well. The biased performance estimate on the training set is significantly bigger than other estimates. For 100 and more observations, BBC SL method leads to the results very close to the nested cross validation. The results of the independent CV are more unstable, often overestimated.

These results are, however, ambiguous: the bias of cross validation methods due to the smaller size of training sets seems to be stronger than any bias due to overfitting. Nevertheless, the proposed BBC SL algorithm proved better than

any other simple verification algorithm and close to the “gold standard” nested cross validation.

3.2 Biomedical Data

The results of super learner procedure obtained for real data are displayed in Tables 3, 4. The direct measurement of performance for unseen data is impossible in this case, hence a nested cross-validation is a reference. The estimated standard deviation of the distribution of results is shown for each data set. The error of the mean value is smaller, but it is hard to estimate, since the measurements are not mutually independent.

As for artificial data, the performance of Random Forest for combining the base results was very poor, and the non-negative logistic regression performed very close to NNLS method. Thus, both were not shown in the tables.

For all the tested data sets ensemble methods outperform the best single classifier. However, in all the cases the best-performing method was a simple average over all the base results. The linear model and k -best proved nearly as good in some cases. This result is obviously not a rule, our artificial data sets give an example for better performance of linear combinations of classifiers.

Table 2. Artificial data – AUC of super learning for 6 diverse methods for three sizes of artificial data sets.

SL method	Training set	Independent CV	BBC SL	Nested CV	New data 100%	New data 90%
50 observations, MSE=0.01						
NNLS	0.66	0.62	0.62	0.59	0.64	0.62
NNlog	0.65	0.62	0.62	0.59	0.59	0.58
RF	0.55	0.57	0.56	0.52	0.56	0.57
Best k	0.67	0.64	0.63	0.61	0.63	0.62
Mean	0.63	0.63	0.63	0.63	0.64	0.62
Best 1	0.67	0.64	0.63	0.61	0.64	0.63
100 observations, MSE=0.01						
NNLS	0.71	0.69	0.69	0.68	0.71	0.70
NNlog	0.71	0.68	0.69	0.68	0.70	0.69
RF	0.65	0.66	0.64	0.63	0.66	0.64
Best k	0.73	0.70	0.69	0.68	0.71	0.70
Mean	0.70	0.70	0.70	0.70	0.72	0.70
Best 1	0.72	0.70	0.68	0.67	0.71	0.69
200 observations, MSE=0.005 for Best 1 and Best k , 0.004 for others						
NNLS	0.829	0.815	0.810	0.814	0.851	0.827
NNlog	0.831	0.816	0.805	0.804	0.854	0.828
RF	0.800	0.802	0.790	0.792	0.832	0.808
Best k	0.830	0.812	0.803	0.799	0.852	0.824
Mean	0.808	0.808	0.807	0.808	0.834	0.816
Best 1	0.822	0.807	0.797	0.794	0.851	0.816

Another interesting point is the stability of the results. The most repeatable values of AUC are produced by mean of all the classifiers, while choice of the best single one and mean of k best ones are the most unstable. The effect is not visible on the training set, but clearly appears when nested CV or BBC SL verification methods are used (see especially Table 4).

The performance measured using the proposed BBC SL method is almost exactly the same as obtained by nested cross validation. Two simpler methods of the performance estimation report inflated results.

Figure 1 shows the performance of three combining methods for the BRCA RNA data set in more detailed way. The “Best single” method is simply a common practice of choosing the best-performing classifier to this particular purpose. In a simple cross validation, the result seems to be better than any ensemble model. However, when the external validation is applied, the performance drops significantly and becomes unstable. It turns out, that as simple operation as choice between six classification algorithms is a considerable source of overfitting. Both nested cross validation and the proposed BBC SL method show this clearly. For this particular data set, the optimal strategy is the average of all the base predictions.

Table 3. The performance of super learning for 4 methods for three Neuroblastoma data sets. Random Forest and NNlog methods were omitted. Note the mean square error for **Best 1** and **Best k** methods, bigger than the typical values.

SL method	Training set	Independent CV	BBC SL	Nested CV
CNV, typical MSE=0.02				
NNLS	0.78	0.77	0.76	0.76
Best k	0.78	0.77	0.75	0.76 ± 0.03
Mean	0.76	0.76	0.76	0.76
Best 1	0.78	0.77 ± 0.03	0.74 ± 0.03	0.75 ± 0.03
MA, typical MSE=0.01				
NNLS	0.89	0.87	0.87	0.88
Best k	0.89	0.88 ± 0.02	0.87 ± 0.02	0.88 ± 0.02
Mean	0.89	0.89	0.89	0.89
Best 1	0.89	0.88 ± 0.02	0.85 ± 0.02	0.87 ± 0.02
RNA, typical MSE=0.01				
NNLS	0.91	0.90	0.89	0.89
Best k	0.91	0.90	0.89 ± 0.02	0.89 ± 0.02
Mean	0.89	0.89	0.89	0.89
Best 1	0.90	0.90 ± 0.02	0.88 ± 0.02	0.88 ± 0.02

Table 4. Real data (2) – the performance of super learning for 4 different methods for TCGA data sets. Note the dependence of the square error on the combining method (the biggest for **Best 1** and **Best k**) and on the verification method (smaller on the training set)

SL method	Training set	Independent CV	BBC SL	Nested CV
BRCA RNA				
NNLS	0.9994 ± 0.0001	0.9992 ± 0.0002	0.9991 ± 0.0002	0.9991 ± 0.0002
Best k	0.9994 ± 0.0001	0.998 ± 0.002	0.998 ± 0.001	0.998 ± 0.003
Mean	0.9993 ± 0.0001	0.9993 ± 0.0001	0.9993 ± 0.0001	0.9993 ± 0.0001
Best 1	0.9994 ± 0.0001	0.998 ± 0.003	0.997 ± 0.002	0.996 ± 0.004
BRCA CNV				
NNLS	0.988 ± 0.001	0.987 ± 0.001	0.987 ± 0.001	0.987 ± 0.001
Best k	0.988 ± 0.001	0.987 ± 0.001	0.987 ± 0.001	0.987 ± 0.001
Mean	0.988 ± 0.001	0.988 ± 0.001	0.988 ± 0.001	0.988 ± 0.001
Best 1	0.983 ± 0.001	0.981 ± 0.003	0.981 ± 0.002	0.981 ± 0.002
HNSC				
NNLS	0.996 ± 0.002	0.992 ± 0.005	0.990 ± 0.002	0.993 ± 0.006
Best k	0.997 ± 0.001	0.988 ± 0.01	0.986 ± 0.005	0.98 ± 0.01
Mean	0.995 ± 0.003	0.995 ± 0.003	0.995 ± 0.002	0.995 ± 0.003
Best 1	0.997 ± 0.001	0.985 ± 0.01	0.983 ± 0.006	0.98 ± 0.01
LUAD				
NNLS	0.9993 ± 0.0002	0.998 ± 0.002	0.9978 ± 0.0007	0.998 ± 0.002
Best k	0.9994 ± 0.0001	0.998 ± 0.003	0.997 ± 0.002	0.997 ± 0.003
Mean	0.9992 ± 0.0001	0.9992 ± 0.0001	0.9991 ± 0.0002	0.9992 ± 0.0001
Best 1	0.9993 ± 0.0001	0.998 ± 0.004	0.997 ± 0.002	0.997 ± 0.003

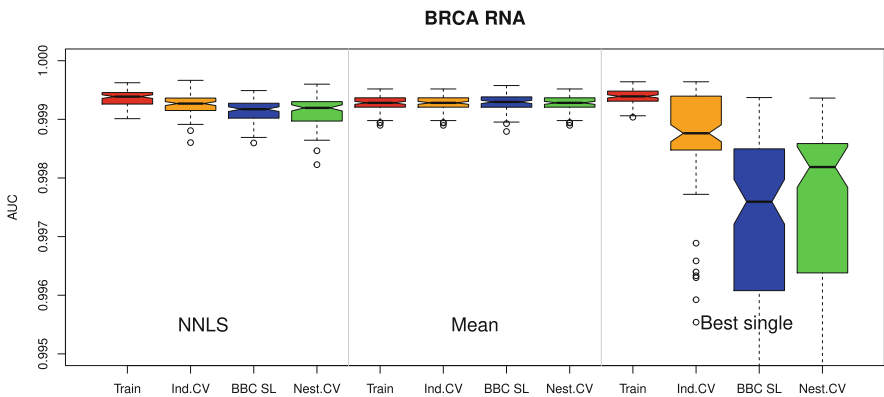


Fig. 1. Box plots of AUC for an example real data set. Note the drop of performance and the instability of results for the best single classifier, when unbiased verification methods are applied.

4 Conclusion

Main Contribution. Super learner as proposed in the [3] is computationally demanding approach that relies on multiple cross-validation and application of multiple learning algorithms. In the original formulation of the algorithm, verification of the quality of the final model involves repeating entire procedure within nested cross-validation, what significantly increases the computational cost of the modelling procedure. The current study shows that the nested cross-validation can be replaced by using resampling protocol, which gives equivalent results.

Additional Remarks. In almost all examined cases, the ensemble of learners gives better results than selection of a single best method for prediction. Interestingly, the simplest method of combining results of all algorithms, namely the the simple average of predictions of all base learners seems to be a very good choice for obtaining a stable non-overfitted estimate. Two simplest methods for assigning weights to base learners, namely a simple linear combination or selection of unweighted average of k-best models have better performance than other methods and should be explored along simple mean of all base methods.

One should note, that Super learner was applied here to merge results of algorithms that are very good predictors themselves. The differences between predictions of these classifiers are concentrated on a handful of difficult cases. The simple average works best, because there are too few independent data point to build reliable model for more advanced methods.

References

1. Fernández-Delgado, M., Cernadas, E., et al.: Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* **15**(1), 3133–3181 (2014)
2. Marbach, D., Costello, J.C., et al.: Wisdom of crowds for robust gene network inference. *Nat. Methods* **9**(8), 796 (2012)
3. Van der Laan, M.J., Polley, E.C., et al.: Super learner. *Stat. Appl. Genet. Mol. Biol.* **6**(1) (2007)
4. Saez-Rodriguez, J., et al.: Crowdsourcing biomedical research: leveraging communities as innovation engines. *Nat. Rev. Genet.* **17**(8), 470 (2016)
5. Tsamardinos, I., Greasidou, E., Borboudakis, G.: Bootstrapping the out-of-sample predictions for efficient and accurate cross-validation. *Mach. Learn.* **107**(12), 1895–1922 (2018). <https://doi.org/10.1007/s10994-018-5714-4>
6. Efron, B., Tibshirani, R.J.: *An Introduction to the Bootstrap*. CRC Press, New York (1994)
7. Zhang, W., Yu, Y., et al.: Comparison of RNA-seq and microarray-based models for clinical endpoint prediction. *Genome Biol.* **16**, 133 (2015). <https://europepmc.org/article/PMC/4506430>
8. Theissen, J., Oberthuer, A., et al.: Chromosome 17/17q gain and unaltered profiles in high resolution array-CGH are prognostically informative in neuroblastoma. *Genes Chromos. Cancer* **53**(8), 639–649 (2014). <https://onlinelibrary.wiley.com/doi/abs/10.1002/gcc.22174>

9. Stigliani, S., Coco, S., et al.: High genomic instability predicts survival in metastatic high-risk neuroblastoma. *Neoplasia* **14**(9), 823–910 (2012). <http://www.science-direct.com/science/article/pii/S1476558612800868>
10. Coco, S., Theissen, J., et al.: Age-dependent accumulation of genomic aberrations and deregulation of cell cycle and telomerase genes in metastatic neuroblastoma. *Int. J. Cancer* **131**(7), 1591–1600 (2012). <https://onlinelibrary.wiley.com/doi/abs/10.1002/ijc.27432>
11. Kocak, H., Ackermann, S., et al.: Hox-C9 activates the intrinsic pathway of apoptosis and is associated with spontaneous regression in neuroblastoma. *Cell Death Dis.* **4**, e586 (2013)
12. Edgar, R., Domrachev, M., et al.: Gene expression omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.* **30**(1), 207–210 (2002)
13. Hammerman, P., Lawrence, M., et al.: Comprehensive genomic characterization of squamous cell lung cancers. *Nature* **489**, 519–525 (2012)
14. Collisson, E., Campbell, J., et al.: Comprehensive molecular profiling of lung adenocarcinoma. *Nature* **511**, 543–550 (2014)
15. Koboldt, D., Fulton, R., et al.: Comprehensive molecular portraits of human breast tumours. *Nature* **490**, 61–70 (2014)
16. Ciriello, G., Gatza, M.L., et al.: Comprehensive molecular portraits of invasive lobular breast cancer. *Cell* **163**(2), P506–519 (2015)
17. Lawrence, M., Sougnez, C., et al.: Comprehensive genomic characterization of head and neck squamous cell carcinomas. *Nature* **517**, 576–582 (2015)
18. Creighton, C., Morgan, M., et al.: Comprehensive molecular characterization of clear cell renal cell carcinoma. *Nature* **499**, 43–49 (2013)
19. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
20. Friedman, J., Hastie, T., et al.: Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.* **33**(1), 1–22 (2010)
21. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol. (TIST)* **2**(3), 1–27 (2011)
22. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P. (ed.) *EuroCOLT 1995*. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-59119-2_166