ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Dynamic Ensemble Selection and Instantaneous Pruning for Regression

Kaushala Dias and Terry Windeatt

Centre for Vision Speech and Signal Processing
Faculty of Engineering and Physical Sciences
University of Surrey, Guildford, Surrey, GU2 7XH
United Kingdom

Abstract. A novel dynamic method of selecting pruned ensembles of predictors for regression problems is presented. The proposed method, known henceforth as DESIP, enhances the prediction accuracy and generalization ability of pruning methods. Pruning heuristics attempt to combine accurate yet complementary members, therefore DESIP enhances the performance by modifying the pruned aggregation through distributing the ensemble member selection over the entire dataset. Four static ensemble pruning approaches used in regression are compared to highlight the performance improvement yielded by the dynamic method. Experimental comparison is made using Multiple Layer Perceptron predictors on benchmark datasets.

1 Introduction

In the context of ensemble methods, it is recognized that the combined outputs of several regressors generally give improved accuracy compared to a single predictor [1]. It has also been shown that ensemble members that are complementary can be selected to further improve the performance [1]. The selection, also called pruning, has the potential advantage of both reduced ensemble size as well as improved accuracy. However the selection of classifiers, rather than regressors, has previously received more attention and given rise to many different approaches to pruning [2]. Some of these methods have been adapted to the regression problem [3]. The dynamic pruning methods in [4, 5] are classification oriented and rely on functions that determine the ensemble selection based on information from the training set. The proposed novel dynamic method for regression similarly uses the training set to determine the ensemble selection. By dynamic, we mean that the subset of predictors is chosen differently depending on the test sample and its relationship to the training set.

2 Related Research

The main objective of using ensemble methods in regression problems is to harness the complementarity of individual ensemble member predictions [1]. In Negative Correlation Learning, diversity of the predictors is introduced by simultaneously training a collection of predictors using a cost function that includes a correlation penalty term [3]; thereby collectively enhancing the performance of the entire ensemble. By weighting the outputs of the ensemble members before aggregating, an

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

optimal set of weights is obtained in [9] by minimizing a function that estimates the generalization error of the ensemble; this optimization being achieved using genetic algorithms. With this approach, predictors with weights below a certain level are removed from the ensemble. In [3] genetic algorithms have been utilized to extract sub-ensembles from larger ensembles. In Stacked Generalization a meta-learner is trained with the outputs of each predictor to produce the final output [1]. Empirical evidence shows that this approach tends to over-fit, but with regularization techniques for pruning ensembles over-fitting is eliminated. Ensemble pruning by Semi-definite Programming has been used to find a sub-optimal ensemble in [3]. A dynamic ensemble selection approach in which many ensembles that perform well on an optimization set or a validation set are searched from a pool of over-produced ensembles and from this the best ensemble is selected using a selection function for computing the final output for the test sample [4]. Similarly a dynamic multistage organizational method based on contextual information of the training data is used to select the best ensemble for classification in [5]. Recursive Feature Elimination has been used in [7] as a method of pruning ensembles. Here the weights of a trained combiner are evaluated to determine the least performing predictor that is removed from the ensemble.

2.1 **Reduced Error Pruning**

Reduced Error Pruning without back fitting method (RE) [2], modified for regression problems, is used to establish the order of regressors in the ensemble that produces a minimum in the ensemble training error. Starting with the regressor that produces the lowest training error, the remaining regressors are subsequently incorporated one at a time into the ensemble to achieve a minimum ensemble error. The sub ensemble S_u is constructed by incorporating to S_{u-1} the regressor that minimizes

$$s_u = \arg_k \min u^{-1}\left(\sum_{i=1}^{u-1} C_{s_i} + C_k\right) \tag{1}$$

where $k \in (1,...,M)\backslash\{S_1, S_2,\ldots,S_{u-1}\}$ and $\{S_1, S_2,\ldots,S_{u-1}\}$ label regressors that have been incorporated in the pruned ensemble at iteration $u-1$. For static ensemble selection C_i is calculated over the entire training set and expressed as

$$C_i = \sum_{n=1}^{N} f_i(x_n) - y_n \tag{2}$$

where $i = 1,2,\ldots,M$, $f_i(x)$ is the output of the i^{th} regressor and(x_n, y_n) is the training data where $n = (1,2,\ldots,N)$. Therefore the information required for the optimization of the training error is contained in the vector C.

3 Method

In contrast to static ensemble selection, Dynamic Ensemble Selection with Instantaneous Pruning (DESIP) provides an ensemble tailored to the specific test instance based on the information of the training set. The method described here is for a regression problem where the regressors are ordered for every individual training instance based on the method of RE. Therefore each ensemble selection for every

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

training instance contains the same regressors as constituent members but aggregated in a different order. However, potentially this dynamic method can be implemented with any pruning technique.

The implementation of DESIP consists of two stages. First the base regressors M are trained on bootstrap samples of the training dataset and the regressor order is found for every instance in the training set. As shown in the pseudo-code in figure 1, this is achieved by building a series of nested ensembles, per training instance, in which the ensemble of size u contains the ensemble of size u-1. Taking a single instance of the training set, the method starts with an empty ensemble *S*, in step 2, and builds the ensemble order, in steps 6 to 15, by evaluating the training error of each regressor in M. The regressor that increases the ensemble training error least is iteratively added to *S*. This is achieved by minimizing *z* in step 9. Therefore each regressor in M takes a unique position in *S* as *S* grows. This order is archived in a two dimensional matrix *A* with regressor order in rows and training instance in columns.

Training data $D = (x_n, y_n)$, where $n = (1,2,..,N)$ and f_m is a regressor,
where $m = (1,2,..,M)$. The Archive Matrix $A = (a_n)$ where a_n is a column vector with max index of *m*. *S* is also a vector with max index of *m*.

1. **For** $n = 1….N$
2. $S \leftarrow$ empty vector
3. **For** $m = 1…M$
4. Evaluate $C_m = f_m(x_n) - y_n$
5. **End for**
6. **For** $u = 1…M$
7. min $\leftarrow +\infty$
8. **For** k in $(1,...,M)\backslash\{S_1, S_2,…,S_u\}$
9. Evaluate $z = u^{-1}\left(\sum_{i=1}^{u-1} C_{S_i} + C_k\right)$
10. **If** $z <$ min
11. $S_u \leftarrow k$
12. min $\leftarrow z$
13. **End if**
14. **End for**
15. **End for**
16. $a_n \leftarrow S$
17. **End for**

Fig 1: Pseudo-code implementing the archive matrix with ordered ensemble per training instance.

In the second stage, the regressor order that is associated with the training instance closest to the test instance is retrieved from matrix *A*. Here the closeness is determined by calculating the L1 Norm of the distance measure between the test instance and the training set. This is performed in steps 1 to 6 in figure 2. All input features of the training set are considered to identify the closest training instance,

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

using the K-Nearest Neighbors method [6], where K = 1. The resulting vector g_n, where n is the index of the training instance, is searched for the minimum value and is identified as the closest training instance to be retrieved from A. The selected ensemble has the order of regressors determined by the training instance.

Test and train instance $x_{f,test}$, $x_{f,n,train}$ where f = (1,2,..,F) features.
From figure 1 Archive Matrix $A = (a_n)$ where a_n is the column vector containing the order of regressors, n = 1,2,..,N.
e_f is a vector with max index of F and g_n is a vector with max index of N.

1. **For** n = 1….N
2. **For** f = 1….F
3. Evaluate $e_f = |x_{f,n,train} - x_{f,test}|$
4. **End for**
5. $$g_n = \left(\sum_{f=1}^{F} e_f \right)$$
6. **End for**
7. Search for the minimum values in g_n and note n
8. a_n is the ensemble selection for the test instance.

Fig 2: Pseudo-code implementing the identification of the closest training instance to the test instance.

In the implementation of DESIP with RE, equation (2) is modified so that C_i is calculated for every training instance. The modified equation is shown in equation (3)

$$C_i = f_i(x_n) - y_n \tag{3}$$

Consequently S_u in equation (1) is also calculated for every training instance.

For the comparison of DESIP with static methods, four static pruning methods were implemented with DESIP. They are Ordered Aggregation (OA) as described in [3] for regression, Recursive Feature Elimination (RFE) in [7], ensemble optimization using Genetic Algorithm (GA) [3] and Reduced Error Pruning without back fitting (RE), described in Section 2.1. The datasets listed in table 3 have been used for the above comparison.

4 Results

MLP architecture using the Levenberg Marquardt learning algorithm with 5 nodes in the hidden layer, as described in [3] has been selected in this experiment. The training/test data split is 70/30 percent, and 32 base regressors are trained with bootstrap samples. The Mean Squared Error (MSE) is used as the performance indicator for both training and test sets, and averaged over 100 iterations.

Tables 1 and 2 show MSE performance of the four static methods with and without DESIP. In tables 1 and 2, grayed results indicate the minimum MSE over the eight methods. It is observed that the majority of the lowest MSE values have been achieved by DESIP. Figure 3 shows the comparison of the training and the test error

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

plots of static methods and DESIP (with RE implemented) for the SERVO dataset. It is observed that pruned ensembles with DESIP are more accurate with fewer members than static methods.

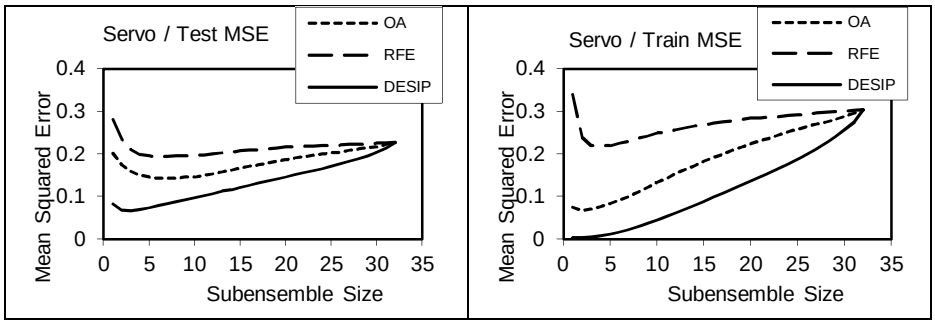

Fig 3: Comparison of the MSE plots of the training set and the test set for OA, RFE and DESIP using RE.

Dataset	Multiplier	OA	RFE	GA	RE
Servo	10^{-1}	1.43±0.40	1.94±0.50	2.37±0.4	1.42±0.24
Boston Housing	10^{0}	7.94±0.60	9.12±0.94	9.91±0.83	7.97±0.64
Forest Fires	10^{0}	1.78±0.05	1.78±0.08	1.81±0.18	1.78±0.07
Wisconsin	10^{1}	2.54±0.10	2.55±0.10	2.58±0.17	2.53±0.12
Concrete Slump	10^{1}	3.02±0.16	3.54±0.26	3.66±0.39	2.89±0.36
Auto93	10^{1}	5.50±0.81	6.37±0.77	6.52±0.66	5.58±0.93
Auto Price	10^{6}	3.81±0.64	5.04±1.89	6.09±8.03	3.88±0.63
Body Fat	10^{-1}	6.91±1.23	6.80±6.70	8.44±2.81	5.99±2.12
Bolts	10^{1}	7.70±3.90	10.50±3.71	10.95±3.71	7.81±4.69
Pollution	10^{3}	2.28±0.22	2.47±0.22	2.54±0.31	2.26±0.21
Sensory	10^{-1}	8.70±0.30	6.10±0.20	6.29±0.19	5.73±0.26

Table 1: Static Ensemble Pruning Methods: Averaged MSE with Standard Deviation for the 100 iterations.

Dataset	Multiplier	OA	RFE	GA	RE
Servo	10^{-1}	0.66±0.21	1.46±0.43	1.35±0.30	0.66±0.21
Boston Housing	10^{0}	6.47±0.78	8.70±0.86	8.06±0.77	6.47±0.78
Forest Fires	10^{0}	1.72±0.08	1.76±0.10	1.79±0.76	1.72±0.09
Wisconsin	10^{1}	2.27±0.28	2.29±0.08	2.18±0.14	2.27±0.28
Concrete Slump	10^{1}	3.09±0.56	3.16±0.23	3.14±0.39	3.09±0.56
Auto93	10^{1}	5.85±0.83	6.35±0.81	6.51±0.67	5.85±0.83
Auto Price	10^{6}	3.88±0.63	5.16±2.13	5.85±4.59	3.88±0.63
Body Fat	10^{-1}	6.12±1.57	6.17±3.10	5.96±1.50	6.12±1.57
Bolts	10^{1}	7.44±2.43	7.42±2.82	7.33±2.55	7.44±2.43
Pollution	10^{3}	2.15±0.26	2.30±0.19	2.35±0.30	2.15±0.26
Sensory	10^{-1}	5.48±0.16	5.85±0.13	6.11±0.20	5.48±0.16

Table 2: DESIP with Static Methods Adopted: Averaged MSE with Standard Deviation for the 100 iterations.

Dataset	Instances	Attributes	Source
Servo	167	5	UCI-Repository
Boston Housing	506	14	UCI-Repository
Forest Fires	517	14	UCI-Repository
Wisconsin	198	36	UCI-Repository
Concrete Slump	103	8	UCI-Repository
Auto93	82	20	WEKA
Auto Price	159	16	WEKA
Body Fat	252	15	WEKA
Bolts	40	8	WEKA
Pollution	60	16	WEKA
Sensory	576	12	WEKA

Table 3: Benchmark datasets used

5 Conclusion

Dynamic ensemble pruning utilizes a distributed approach to ensemble selection and is an active area of research for both classification and regression problems. In this paper, a novel method of dynamic pruning of regression ensembles is proposed. Experimental results show that test error has been reduced by modifying the pruning based on the closest training instance. On a few datasets the proposed method has not improved performance, and will be investigated further along with different distance measures, varying K for K-NN and relevant feature selection. Bias/Variance and time complexity analysis should also help to understand the performance relative to other static and dynamic pruning methods with similar complexity.

References

[1] Tsoumakas G., Partalas I., Vlahavas I., An Ensemble Pruning Primer. Supervised and Unsupervised Ensemble Methods and their Applications. Studies in Computational Intelligence Volume 245, Springer 2009, pp 1 – 13.

[2] Martínez-Muñoz G., Hernández-Lobato D., Suárez A., An Analysis of Ensemble Pruning Techniques Based on Ordered Aggregation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 31(2), 2009. pp 245 – 259.

[3] Hernández-Lobato D., Martínez-Muñoz G., Suárez A, Empirical Analysis and Evaluation of Approximate Techniques for Pruning Regression Bagging Ensembles. Neurocomputing 74, 2011, pp 2250 – 2264.

[4] Dos Santos E.M., Sabourin R., Maupin P., A Dynamic Overproduce-and-choose Strategy for the selection of Classifier Ensembles. Pattern Recognition 41, 2008, pp 2993 – 3009.

[5] Cavalin P.R., Sabourin R., Suen C.Y., Dynamic Selection of Ensembles of Classifier Using Contextual Information, Multiple Classifier Systems Volume 5997, LNCS, Springer, 2010, pp 145 – 154.

[6] Dubey H., Pudi V., CLUEKR: Clustering Based Efficient K-NN Regression, Advances in Knowledge Discovery and Data Mining, LNCS, Springer, Volume 7818, 2013, pp 450 – 458.

[7] Windeatt T., Dias K., Feature Ranking Ensembles for Facial Action Unit Classification. IAPR Third International Workshop on Artificial Neural Networks in Pattern Recognition, 2008.

[8] Cavalin P.R., Sabourin R., Suen C.Y., Dynamic Selection Approaches for Multiple Classifier Systems, Neural Computing Applications Volume 22 (3-4) LNCS, Springer, 2013, pp 673 – 688.

[9] Zhau Z.-H., Wu J., Tang W., Ensembling Neural Networks: many could be better than all, Artificial Intelligence, Volume 137, 2002, pp 239 – 263.