ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

NNBMSS: a Novel and Fast Method for Model Structure Selection

Amaury Lendasse[1,2], Kallin Khan[3] and Edward Ratner[3]

1- University of Houston - Department of Information and Logistics Technology
Houston - USA

2- Arcada University of Applied Sciences - Risklab
Helsinki - Finland

3- Edammo Inc.
Iowa City - USA

Abstract. In this paper, we present a new method to perform model structure selection. This proposed method can be used to select the complexity of any continuous regression method. We also present an asymptotic mathematical proof of the proposed method and the new method is illustrated on a benchmark. Compared to the well-known 10-fold Cross-Validation, the computational time associated to our new method is approximately divided by a factor 8 as illustrated on the benchmark.

1 Introduction

In machine learning, modeling is the creation and tuning of an algorithm to take in certain data, recognize patterns amongst the data, and produce predictions based on this and future data. The input and output of information to a model is similar to that of a mathematical function. Regression is a modeling technique that utilizes supervised learning, where the expected output of an input is known, in order to train models to predict continuous values from several input variables. Much research has been dedicated to creating new algorithms to perform regression across a broad domain. However, when a new data set arises, it is often difficult to select which particular regression model to use. Further, once a certain algorithm is chosen, tuning the structure for this algorithm can prove to be even more labor and computationally intensive. Therefore in this paper, we propose a new method to perform model structure selection.
Model selection is the process of selecting several different machine learning models, training them each on the same set of training data, and then evaluating the models on their ability to perform predictions on a new and unused set (validation set, [1, 2, 3]). Model structure selection is the process of changing the complexity of a model in order for it to perform better on new data. The difficulty in model structure selection arises due to the fact that if a certain model structure predicts extremely accurately on the training set or even the validation set, then it is not necessarily the best structure to use in future prediction. This is due to the potential for overfitting or underfitting.
Overfitting occurs when a statistical model or machine learning algorithm captures the noise of the data [1]. Intuitively, overfitting occurs when the model or the algorithm fits the data too well. Specifically, overfitting occurs if the model

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

or algorithm shows low bias but high variance. Overfitting is often a result of an excessively complicated model, and it can be prevented by fitting multiple models and using validation or Cross-Validation to compare their predictive accuracies on validation data. Underfitting occurs when a statistical model or machine learning algorithm cannot capture the underlying trend of the data. Intuitively, underfitting occurs when the model or the algorithm does not fit the data well enough. Specifically, underfitting occurs if the model or algorithm shows low variance but high bias. Underfitting is often a result of an excessively simple model. Both overfitting and underfitting lead to poor predictions on new data sets. Ideally, we want to select a model at the sweet spot between underfitting and overfitting, and we will denote such sweet spot goodfitting.
Due to the potential of sampling error in the training and validation sets, a goodfitting model has the lowest generalization error. Generalization error is defined as a quantity for how well a model can predict on data that was not used to train the model. The current state of the art for model structure selection, 10-fold cross validation, is estimating the generalization error. Put simply, 10-fold cross validation is performed on a model by training the model on 90% of the data and validating on the remaining 10%. This process is repeated 10 times, with a completely different 10% of data used in validation and remaining 90% in training each time. The generalization error of the model structure can then be evaluated based on average of the 10 validation set prediction errors. Though the 10-fold cross validation is good at selecting the complexity of a model, the issue is that this method is slow since the model must be retrained 10 times. With the increased complexity and thus increased training times of new machine learning models, using 10-fold cross validation for model structure selection often is not feasible. For this reason, in this paper we propose a new method, called Nearest-Neighbor Based Model Structure Selection (NNBMSS), which can perform several times faster than 10-fold cross validation without a loss in performance. In short, NNBMSS is able to achieve such speed by only training the model once as described further below.
In Section 2, the proposed method is introduced. In Section 3, an asymptotic mathematical proof is presented using the concepts of overfitting, underfitting and goodfitting. In Section 4, the method is illustrated on a benchmark and in Section 5, these results are analysed.

2 The Proposed Method

As stated, our novel method for model structure selection resolves the issue of high computation complexity of the current state of the art, 10-fold cross validation. This new algorithm, called Nearest-Neighbor Based Model Structure Selection (NNBMSS), achieves this through the use of the following metric: $\mathrm{Var}[y_i - \hat{y}_{NN(i)}]$ (defined in Section 3). This metric is a good estimator to evaluate if a regression model is optimal or not, as demonstrated in an experiment in Section 4.
There are two main advantages to NNBMSS. First, NNBMSS uses all available

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

training data to select the complexity of the model. For each step of a 10-fold cross validation, only 90% of the data is used to train the model. This may result in sampling error, whereas NNBMSS, which utilizes all of the data, is less prone to this problem. Though NNBMSS has to search the Nearest Neighbor of each sample to find the optimal model structure, which is computationally expensive, NNBMSS still performs significantly faster than 10-fold cross validation due to NNBMSS's second main advantage: the model is built only once. The proposed method is related to the Delta Test that has been used for model structure selection in [4, 5]. The Delta Test is a non-parametric technique that is utilized to estimate variance in a supervised learning context, such as regression. In the following section, we are going to give an asymptotic mathematical proof of the efficacy of the NNBMSS to select the optimal model structure in the context of a regression problem. Then, in Section 4, we are going to show that NNBMSS works in practice and is significantly faster than the current state of the art.

3 Mathematical Proof

In this proof, the regression model is denoted by f and the samples (points) are denoted (x_i, y_i) with $1 \leqq i \leqq N$, N being the number of samples. For each sample, y_i is the output of the input x_i. Using the model f, the approximation provided by the regression model is denoted $\hat{y}_i$ and we can express the regression model as $\hat{y}_i = f(x_i)$. The nearest neighbor of x_i is denoted $x_{NN(i)}$, therefore the output of $x_{NN(i)}$ is $y_{NN(i)}$. The first assumption is that the output y_i is noisy and therefore is not the optimal approximation for the input x_i. We choose to denote the optimal approximation $\tilde{y}_i$. We can then write $y_i = \tilde{y}_i + \epsilon_i$ with ϵ a random variable with zero mean and variance σ^2. The second assumption is that the input space is bounded and that the true regression problem and its approximation are continuous. Then, for N that tends to infinity, the nearest neighbor of a sample is getting closer and closer to this sample itself, and therefore

$$\hat{y}_{NN(i)} \xrightarrow[N\to\infty]{} \hat{y}_i \tag{1}$$

and also

$$\tilde{y}_{NN(i)} \xrightarrow[N\to\infty]{} \tilde{y}_i. \tag{2}$$

Our hypothesis is that $\mathrm{Var}[y_i - \hat{y}_{NN(i)}]$ is a good estimator to evaluate if a regression model is optimal or not. $\mathrm{Var}[y_i - \hat{y}_{NN(i)}]$ is minimal of the goodfitting state. We are going to divide the proof into 3 possible cases: the model is underfitting, the model is strongly overfitting and the model is optimal.

3.1 Strong Overfitting State

If the regression model is strongly overfitting then $\hat{y}_{NN(i)}$ is tending to $y_{NN(i)}$ and we can expend $y_i - \hat{y}_{NN(i)}$ using (2) as

$$y_i - \hat{y}_{NN(i)} = y_i - y_{NN(i)} = (\tilde{y}_i - \epsilon_i) - (\tilde{y}_{NN(i)} - \epsilon_{NN(i)}) = \epsilon_i - \epsilon_{NN(i)}. \tag{3}$$

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

So, we can calculate the variance of $y_i - \hat{y}_{NN(i)}$ as

$$\mathrm{Var}[y_i - \hat{y}_{NN(i)}] = \mathrm{Var}[\epsilon_i - \epsilon_{NN(i)}] = 2\sigma^2. \tag{4}$$

3.2 Underfitting State

In the case of underfitting, we have that $\hat{y}_i = \tilde{y}_i + \alpha_i$ with α a random variable with zero mean and variance μ^2. It can be rewritten as $\hat{y}_i = y_i - \sigma_i + \alpha_i$ and we can expend $y_i - \hat{y}_{NN(i)}$ as

$$y_i - \hat{y}_{NN(i)} = \hat{y}_i - \alpha_i + \epsilon_i - \hat{y}_{NN(i)}. \tag{5}$$

For the number of samples that tends to infinity and using (1), we can rewrite (5) as

$$y_i - \hat{y}_{NN(i)} = -\alpha_i + \epsilon_i. \tag{6}$$

So, we can calculate the variance of $y_i - \hat{y}_{NN(i)}$ as

$$\mathrm{Var}[y_i - \hat{y}_{NN(i)}] = \mathrm{Var}[\epsilon_i - \alpha_i] = \sigma^2 + \mu^2. \tag{7}$$

3.3 Goodfitting State

When the complexity of the regression model is optimal and if the number of samples that tends to infinity, using (1), we obtain

$$y_i - \hat{y}_{NN(i)} = (\tilde{y}_i + \epsilon_i) - \tilde{y}_{NN(i)} = (\tilde{y}_i - \tilde{y}_{NN(i)}) + \epsilon_i = \epsilon_i. \tag{8}$$

So, we can calculate the variance of $y_i - \hat{y}_{NN(i)}$ as

$$\mathrm{Var}[y_i - \hat{y}_{NN(i)}] = \mathrm{Var}[\epsilon_i] = \sigma^2. \tag{9}$$

3.4 The NNBMSS is minimal for the Goodfitting State

The estimator $\mathrm{Var}[y_i - \hat{y}_{NN(i)}]$ is minimal and tends to σ^2 for the goodfitting state, instead of $2\sigma^2$ for the strong overfitting state and $\sigma^2 + \mu^2$ for the underfitting state. The NNBMSS evolves from σ^2 to $2\sigma^2$ when the model is moving from the goodfitting state through a moderate overfitting state to finally reach a strong overfitting state. Therefore, minimizing $\mathrm{Var}[y_i - \hat{y}_{NN(i)}]$ is a good strategy to choose the complexity of a regression model. □

4 Experiments: NNBMSS VS 10-Fold Cross-Validation

The proposed method (NNBMSS) is tested on a well-known benchmark: the Boston housing dataset. This dataset has been measured to predict the Median value of owner-occupied homes in $1000's [6]. This data consists in 506 samples with thirteen different features including per capita crime rate by town, proportion of residential land zoned for lots over 25,000 sq.ft. and average number of rooms per dwelling. One third of the data is used as a test set in order to calculate the test error of the selected models. This benchmark has been selected

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

because the number is sample is small and therefore far from our asymptotic assumption. For the matter of simplicity, the regression model that is used is a Randomized Neural Networks (RNNs), which is a type of generalized Single-Layer Feed-forward Network (SLFN) [7, 8, 9]. Any other regression model can be substituted to RNNs. For RNNs, the complexity is defined by the number of neurons in the single hidden layer. The proposed model structure selection method (NNBMSS) is compared to 10-fold Cross-Validation [1, 2, 3] which is one of the most used and most robust model structure selection techniques. The selected number of neurons (complexity), the test error (obtained on the test set) and the computational time are compared for the NNBMSS and the 10-fold Cross-Validation. The experiments are repeated 100 times in order to assess the variability of the results. Table 1 is summarizing the results.

Method	Test error	Comp. Time	Selected Number of Neurons
10-fold CV	**34.31** ±5.24	1.28 seconds	**56.65** ±35.35
NNBMSS	**30.24** ±7.02	0.16 seconds	**104.30** ±21.06

Table 1: Results including the means (in bold, calculated over 100 repetitions) and the standard deviations.

5 Discussion: Similar Selection but Faster

The results presented in the previous section show that the NNBSS method is able to select the complexity of a regression model. The results obtained with NNBSS and the 10-fold Cross-Validation are statistically similar, and we are not claiming that the NNBSS method is selecting a better complexity than the 10-fold Cross-Validation. Nevertheless, using 10-fold Cross-Validation is approximately multiplying the computational time by 8.
Exact Nearest Neighbors algorithms have the following complexity: $\mathcal{O}(N \log N)$. In order to really take advantage of the NNBSS method, its utilization should be restricted to regression methods that have similar or worse complexity. Examples of such regression models include Multilayer Perceptrons using back-propagation as part of their training algorithm like Deep Learning models. The complexity of the training using back-propagation is the following : $\mathcal{O}(N^4)$ [10, 11]. Another class of algorithms that can be used (with the NNBMSS method) are Random Forests which have the following training complexity: $\mathcal{O}(N^2)$.
Similar results have been obtained with 5 other regression datasets from the UCI repository. The NNBSS method is selecting similar complexities than the 10-fold Cross-Validation method, and the test errors for both selection method at not statistically different. The NNBSS is on average 8 time faster than the 10-fold Cross-Validation method.

6 Conclusion and Further Work

The proposed method provides the complexity selection for any continuous regression method. An asymptotic mathematical proof is detailed in this paper

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

and the NNBMSS is illustrated on a regression benchmark.
The proposed method is much faster than the 10-fold Cross-Validation method. The bottleneck of the NNBMSS is the search for the Nearest Neighbor of each sample, but we believe that approximate Nearest Neighbors algorithms can be used instead, in order to reduce the complexity and even reduce the computational time even further. In the future, in an extended journal version of this paper, a analysis of the number of samples that is needed for both 10-fold Cross-Validation and NNBMSS methods will be done. Furthermore, more regression models (Radial Basis Function Networks, Deep Learning, Support Vector Machines and Random Forests) will be used on a larger number of datasets. Finally, we will investigate the use of approximate Nearest Neighbor searches (for example Locality Sensitive Hashing [12]) with a very large dataset. It will be determined if approximate Nearest Neighbor searches still select the correct complexity and if they reduce even further the required computational time.

References

[1] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. pages 1137–1143. Morgan Kaufmann, 1995.

[2] Simon Haykin. *Neural Networks: A comprehensive foundation*. Prentice Hall PTR Upper Saddle River, NJ, USA, 2004.

[3] Amaury Lendasse, Vincent Wertz, and Michel Verleysen. Model Selection with Cross-Validations and Bootstraps - Application to Time Series Prediction with RBFN Models. In *Artificial Neural Networks and Neural Information Processing - ICANN/ICONIP 2003. LNCS, vol 2714.*, pages 573–580. Springer, Berlin, Heidelberg, 2003.

[4] Federico Montesino Pouzols, Amaury Lendasse, and Angel Barriga Barros. Autoregressive time series prediction by means of fuzzy inference systems using nonparametric residual variance estimation. *Fuzzy Sets and Systems*, 161(4):471 – 497, 2010.

[5] Elia Liitiäinen, Amaury Lendasse, and Francesco Corona. Non-parametric residual variance estimation in supervised learning. In *Computational and Ambient Intelligence*, pages 63–71, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[6] David Harrison and Daniel L Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81 – 102, 1978.

[7] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 985–990, 2004.

[8] Erik. Cambria et al. Extreme Learning Machines [Trends & Controversies]. *IEEE Intelligent Systems*, 28(6):30–59, Nov 2013.

[9] Yoan Miche, Antti Sorjamaa, Patrick Bas, Olli Simula, Christian Jutten, and Amaury Lendasse. OP-ELM: Optimally pruned extreme learning machine. *IEEE Transactions on Neural Networks*, 21(1):158–162, Jan 2010.

[10] Raúl Rojas. *Neural Networks: A Systematic Introduction*. Springer-Verlag, Berlin, Heidelberg, 1996.

[11] Ozgur Ceyhan. Algorithmic complexities in backpropagation and tropical neural networks, arxiv - 2101.00717, 2021.

[12] Malcolm Slaney and Michael Casey. Locality-sensitive hashing for finding nearest neighbors [lecture notes]. *IEEE Signal Processing Magazine*, 25(2):128–131, 2008.