ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 343-348

A new Meta Machine Learning (MML) method based on combining non-significant different neural networks

Andrés Yáñez Escolano, Joaquín Pizarro Junquera ,
Elisa Guerrero Vázquez, Pedro L. Galindo Riaño,

Universidad de Cádiz, Dpto. Lenguajes y Sist. Informáticos
Grupo de Investigación *"Sistemas Inteligentes de Computación"*
C.A.S.E.M. 11510 – Puerto Real (Cádiz), Spain
{andres.yaniez, joaquin.pizarro, elisa.guerrero, pedro.galindo}@uca.es

Abstract. Model combination provides an alternative to model selection. With a little additional effort we can obtain MML models that improve the generalization capabilities of their individual members. However, it has been recognized that the individual members must be as accurate and diverse as possible. In this paper we present a novel method for building MML models by combining neural networks which are not significantly different from the network selected by some model selection method.

Keywords: meta machine learning method, model selection, regression, multiple comparison procedures, resampling methods, RBF networks

1. Introduction

Let $(x, y) \in X \times Y$ be independent and identically distributed (i.i.d.) random variables such that x takes values in $\Re^m$ and y takes values in $\Re$. Define the regression function as:

$$y = g(x) + e \qquad (1)$$

where e is a stochastic component, commonly taken to be i.i.d. with zero mean and constant variance s^2.

In that context, we consider that X and Y are related by a probabilistic relationship, because generally an element of X does not determine uniquely an element of Y. This can be formalized assuming that an unknown probability distribution $p(x, y) = p(x)\, p(y|x)$ is defined over the set $X \times Y$.

Given n observations $\{(x_i, y_i) \in X \times Y\}_{i=1}^{n}$, we are interested in estimating the regression function, providing an estimator that can be used given any new value of $x \in X$, to predict a value $y \in Y$.

Instead of modeling the statistical dependency between input x and output y using a single estimator $f(x, w^*)$, such that $w^* \in \Omega$ where Ω is an abstract set of parameters, in this paper we propose to improve the results by combining $f(x, w^*)$ and artificial neural networks (ANNs) which *are not significantly different* from it.

The rest of this paper is organized as follows. Our MML method is outlined in section 2. A brief introduction to model selection in order to obtain an ANN which approximates the *true dependency* is presented in section 3. The methodology based

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 343-348

on resampling techniques [13] and multiple comparison procedures (MCPs) [9] to estimate the ANNs which are not significantly different from $f(x,w^*)$ are the topic of section 4. Several empirical techniques for combining the obtained ANNs are proposed in section 5. Our experimental results are presented in section 6 and, finally, our conclusions appear in section 7.

2.- The MML method

There are two disadvantages in training many different candidate networks and selecting the best network discarding the rest [1]. First, all of the effort involved in training the remaining networks is wasted. Second, the generalization performance on the validation set has a random component due to the noise on the data, and so the network which had best performance on the validation set might not be the one with the best performance on new test data.

Combining networks can lead to an effective way of improving in the predictions on new data with a little additional effort. However, this idea has been proved to be true only when the ANNs are fairly accurate, but fairly independent in the errors they make [8], [10]. The ANNs generated by our MML method satisfy both conditions: they are not significantly different from the network with minimum prediction risk (accurate) and they have been created varying their complexity (different errors).

The steps of our method may be outlined as follow:

1. Take the whole data set and create *m* resampled data sets.
2. For each resampled set:
   2.1. Train *k* neural networks whose complexity *i* goes from 1 to *k*
   2.2. Test these networks and obtain *k* validation error measures
3. Estimate the class (S_i) with minimum prediction risk, that is the class with minimum validation error mean.
4. Obtain a subset with the network classes which are not significantly different from the network class with minimum prediction risk.
5. For each class, select one member $f_i(x,w^*)$ whose parameter vector w^* minimizes the empirical risk
6. Combine these networks.

In the next sections, these steps will be described in more detail.

3. Model selection

The task of model selection is to choose a functional form from a number of possible competing alternatives, and to estimate the parameters in a way that satisfies a fitness criterion [15]. This criterion, which is called expected risk or risk functional, can be defined as:

$$\boldsymbol{R}(w) = \int_y \int_x \boldsymbol{L}(y, f(x,w))\, p(x, y)\, dx\, dy \qquad (2)$$

where $\boldsymbol{L}(y, f(x,w))$ is the loss function measuring the error.

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 343-348

But, the optimal model cannot be found in practice, because the probability distribution $p(x, y)$ is usually unknown and only a sample of it is available. To overcome this shortcoming we need an induction principle, a general prescription to obtain an estimate $f(x, w^*)$.

The Minimum Prediction Risk (MPR) principle [15] performs a guided search in specification space. Network classes are ordered according to their complexity, forming a nested structure, $S_1 \subset S_2 \subset ... \subset S_k$, where k might be the number of hidden units for MLPs with one hidden layer or the number of centers for RBF networks. Then, from each class S_i, we select one member $f_i(x,w^*)$ whose parameter vector w^* is estimated minimizing the empirical risk:

$$\boldsymbol{R_n}(w) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{L}(y_i, f(x_i,w)) \qquad (3)$$

Later, we estimate the prediction risk associated with $f_i(x, w^*)$ by means of algebraic [12] or resampling methods. Finally, we choose the model with minimum prediction risk.

4. Non-significant different models

As suggested in section 2, in order to improve the generalization capabilities of our MML method, diverse individual ANNs should be generated so as to perform reasonably well on test data. This can be accomplished by the following procedure:

1. Apply Nemenyi test to obtain the network classes which are not significantly different from the network class with minimum prediction risk.
2. Apply the omnibus tests: Repeated Measures ANOVA test, if its assumptions are met or Friedman test in different case.
3. If the global null hypothesis is true (that is, all network classes of this set are not significantly different), finish the process.
4. If the global null hypothesis is false, apply more powerful multiple comparison procedures (t or Wilcoxon paired tests with Bonferroni method for p-values adjustment) and obtain a subset with the network classes which are not significantly different from the network class with minimum prediction risk.

At this point, some remarks about the above method should be done. First, all tests [16] have been applied using a level of signficance $\alpha = 5\%$. Second, when omnibus tests (step 2) are significant, it indicates that at least two of the network classes are significantly different, but not which are. At this point, multiple comparison procedures, which are usually less powerful, are applied (step 4). Third, Nemenyi test (step 1) is a medium power multiple comparison procedure. It may even accept network classes that should be rejected. It is a good procedure to generate an initial but not definitive set of *non-significant* network classes. And finally, the results may improve with a large number of resampled sets: resampling methods estimate better the prediction risk and parametric tests [4], which are more powerful, may be applied on steps 2 and 4. We suggest $m \geq 30$.

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 343-348

Test	Brief description	Assumptions
Friedman test	High power. Omnibus. Nonparametric. Used to compare k > 2 samples.	None
Nemenyi test	Medium power. All pairwise test. Nonparametric. Used to compare k > 2 samples.	None
Repeated mesaures ANOVA	High power. Omnibus. Parametric. Used to compare k > 2 samples.	Normality, compound symmetry / sphericity
t paired test	High power. Parametric. Used to compare k = 2 samples.	Normality
Wilcoxon matched pairs test	High power. Nonparametric. Used to compare k = 2 samples.	None

Table 1: Statistical tests for related samples.

5. Combining networks

Once a set of ANNs has been generated, they must be combined. Most methods are reduced to a linear weighted combination of models in regression (eg. Bagging [2], AdaBoosting [7], [5] and Stacked regression [3]).

In this section, we propose three empirical rules for assigning values to these weights. First, the simplest one, as all networks make *similar* errors, an unweighted average is computed. Second, their weights are proportional to the number of times that each network has been selected as the network with minimum validation error (see section 2, step 2). And third, the weights take values inversely proportional to validation error sum per network.

6. Experimental results

A number of simulations have been conducted to evaluate the efficiency of MML method using RBF networks. The width of the basis functions has been set to $\left\| max(x_i - x_j) \right\| / \sqrt{2n}$ where n is the number of kernels and the highest complexity (k) is 20.

One thousand data sets of several sample sizes (5, 10, 15, 25, 50, 100 and 500 examples) have been generated according to the following experimental functions:

$$y = -0.2x^4 + 1.5x^3 - 6x + 3 + e \ , \ x \in (-2,+2) \tag{4}$$

$$y = 10sin(2x + 6) + e \ , \ x \in (-2,+2) \tag{5}$$

where e is gaussian noise with zero mean and variance equal to twenty per cent of generalization sample standard deviation.

We trained each network with every generated data set, and estimated the generalization errors applying a new large size test set (10000 examples unseen previously).

In order to compare the performance of our MML models with the minimum prediction risk model, we define the observed efficiency of a model m_i as the ratio

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 343-348

that compares the generalization error between the best model and the model m_i. Thus observed efficiency ranges from 0 to 1.

Two resampling techniques have been used in our experiments: bootstraping and random hold-out (with 2/3 of examples in the resampled training sets) [6], [13], [14]. And 50 resampled data sets have been generated per each data set.

Finally, the quadratic loss has been used as loss function in (3).

Tables 2 and 3 show observed efficiency means. The four first columns contain the results of comparing the three strategies for combining ANNs proposed in section 5 (columns from (1) to (3)) with the model with minimum prediction risk selected by the resampling method (column (4)). In column (5) appears the observed efficiency mean between the model with minimum generalization error and the model selected by resampling.

	bootstrap					**random hold-out**				
sample size	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**
5	0.7871	0.8870	0.7121	0.6727	0.3495	0.9015	0.9045	0.8589	0.8073	0.4889
10	0.9021	0.9346	0.7797	0.7507	0.4831	0.9243	0.9377	0.8526	0.7928	0.5545
15	0.9281	0.9383	0.8398	0.7852	0.6003	0.9420	0.9409	0.8725	0.8193	0.6266
25	0.9586	0.9574	0.9212	0.8478	0.7569	0.9605	0.9528	0.9359	0.8684	0.7810
50	0.9736	0.9694	0.9779	0.9522	0.9200	0.9714	0.9682	0.9773	0.9612	0.9285
100	0.9858	0.9871	0.9884	0.9908	0.9776	0.9849	0.9857	0.9881	0.9897	0.9762
500	0.9987	0.9987	0.9987	0.9970	0.9945	0.9983	0.9985	0.9983	0.9969	0.9943

Table 2: RBF networks trained with samples generated from function (4).

	bootstrap					**random hold-out**				
sample size	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**
5	0.7888	0.8332	0.7390	0.7329	0.4935	0.9189	0.9185	0.8991	0.8394	0.4591
10	0.8430	0.8976	0.8361	0.8220	0.6865	0.8185	0.9275	0.9148	0.8877	0.6827
15	0.9189	0.9426	0.9138	0.9119	0.8602	0.9268	0.9368	0.9327	0.9375	0.8863
25	0.9508	0.9542	0.9752	0.9909	0.9722	0.9456	0.9495	0.9700	0.9826	0.9602
50	0.9795	0.9822	0.9849	0.9912	0.9776	0.9771	0.9806	0.9839	0.9899	0.9742
100	0.9892	0.9927	0.9902	0.9941	0.9877	0.9881	0.9910	0.9899	0.9929	0.9846
500	0.9988	0.9993	0.9988	0.9982	0.9969	0.9983	0.9990	0.9983	0.9982	0.9963

Table 3: RBF networks trained with samples generated from function (5)**.**

The simulations suggest that our MML method gives an ensemble with generalization capabilities better than the best model, the ANN with minimum prediction risk. The more the best model moves away from the minimum generalization error model (see column (5)), the better the observed efficiency mean of our MML method is. Furthermore, we observe that the best proposed strategy for combining models is usually the second, and immediately afterwards the first.

7. Conclusions

In this work we have presented a new MML method based on the combination of non-significant different ANNs which are generated by resampling techniques and MCPs. Simulations with artificial data show that the models generated with this method make good generalization errors.

Future work will address the application of this method to other models and also to study how other more powerful p-value adjustment methods [11] can improve the performance.

References

1. Bishop, C. M.: Neural network for pattern recognition. Clarendon Press-Oxford (1995)
2. Breiman, L. Bagging predictors. Machine Learning, 24 (2), pp. 123-140 (1996)
3. Breiman, L. Stacked regressions. Machine Learning, 24 (1), pp. 49-64 (1996)
4. Don Lehmkuhl, L.: Nonparametric statistics: methods for analyzing data not meeting assumptions required for the application of parametric tests. Journal of prosthetics and orthotics Vol. 8, num. 3, pp.105-113 (1996)
5. Drucker, H. Improving Regressors using Boosting Techniques. Proceeding of the 14th International Conference of Machine Learning, pp. 107-115 (1997).
6. Efron, B., Tibshirani, R.: Introduction to the bootstrap, Chapman & Hall (1993)
7. Freund, Y., Schapire, R. E. Experiments with a new boosting algorithm. Proceedings of the 13th Intenational Conference on Machine Learning, pp. 148-156 (1996)
8. Hansen, L. K., Salamon, P. Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(10), pp. 993-1001 (1990).
9. Hsu, J.C. Multiple comparisons. Theory and methods, Chapman&Hall. (1996)
10. Krogh, A., Vedelsby, J. Neural networks ensembles, cross validation and active learning. In Tesauro, G., Touretzky, D. and Leen, T. (Eds.). Advances in Neural Information Processing Systems, vol. 7, pp. 231-238. The MIT Press (1995).
11. Lasarev, M. R.: Methods for p-value adjustment, Oregon Health & Science University, http://medir.ohsu.edu/~geneview/education/dec19_h.pdf (2001).
12. McQuarrie, A., Tsai, C. L. Regression and times series model selection. World Scientific Publishing Co. Pte. Ltd.(1998).
13. Urban Hjorth, J. S. Computer intensive statistical methods (Validation model selection and bootstrap). Chapman & Hall/CRC (1999).
14. Weiss, S. M. and Kulilowski, C. A. Computer Systems That Learn. Morgan Kaufmann (1991)
15. Zapranis, A., Refenes, A. P. Principles of neural model identification, selection and adequacy with applications to financial econometrics. Springer-Verlag (1999).
16. Zar, J. H.: Biostatistical analysis, Prentice Hall (1996)