Untrained Method for Ensemble Pruning and Weighted Combination

Bartosz Krawczyk(✉) and Michał Woźniak

Department of Systems and Computer Networks, Wrocław University
of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{bartosz.krawczyk,michal.wozniak}@pwr.edu.pl

Abstract. The combined classification is an important area of machine learning and there are a plethora of approaches methods for constructing efficient ensembles. The most popular approaches work on the basis of voting aggregation, where the final decision of a compound classifier is a combination of discrete individual classifiers' outputs, i.e., class labels. At the same time, some of the classifiers in the committee do not contribute much to the collective decision and should be discarded. This paper discusses how to design an effective ensemble pruning and combination rule, based on continuous classifier outputs, i.e., support functions. As in many real-life problems we do not have an abundance of training objects, therefore we express our interest in aggregation methods which do not required training. We concentrate on the field of weighted aggregation, with weights depending on classifier and class label. We propose a new untrained method for simultaneous ensemble pruning and weighted combination of support functions with the use of a Gaussian function to assign mentioned above weights. The experimental analysis carried out on the set of benchmark datasets and backed up with a statistical analysis, prove the usefulness of the proposed method, especially when the number of class labels is high.

Keywords: Machine learning · Classifier ensemble · Classifier combination · Ensemble pruning · Weighted fusion · Untrained aggregation

1 Introduction

For a given classification task, we may often have more than a single classifier available. What is interesting, the number of misclassified objects by all individual classifiers is typically small. From this we can conclude, that even if individual classifiers do not have high quality, their union could form a reasonably good compound classifier. The considered approach is called a multiple classifier system (MCS), combined classifier or classifier ensemble and is considered as one of the most vital fields in the contemporary machine learning [10].

During the ensemble design process, we must take into consideration several important aspects, such as which classifiers to use, how to select the proper

Z. Zeng et al. (Eds.): ISNN 2014, LNCS 8866, pp. 358–365, 2014.
DOI: 10.1007/978-3-319-12436-0_40

topology, or what would be the best method for combining their outputs. In this work, we focus on two crucial steps: ensemble pruning and classifier combination.

For most considered problems, we can create / collect a huge number of classifiers. However, for ensemble to work properly it should be formed by mutually complementary models of high individual quality. Adding new classifiers that do not exploit a new area of competence do not improve the ensemble, only increases the computational cost and reduces its robustness. The problem lies on how to select a useful subgroup from a large pool of classifiers at hand.

However, one should note that these methods require specific criteria to evaluate the selected subgroup of classifiers, such as accuracy, AUC or diversity. Such criteria do not often lead to a satisfactory results (as using accuracy may lead to large and similar ensembles, while diversity will not take into account the individual quality of models) and selecting a proper metric for a given problem is not a trivial task.

When having selected a number of competent classifiers, one need to design a combination rule in order to establish a collective decision of the ensemble. Such a mechanism should be able to exploit the individual strengths of classifiers in the pool, while at the same time minimizing their drawbacks. In literature, two methods for classifier combination can be distinguished: methods that make decisions on the basis of discrete outputs (class labels) returned by the individual classifiers and methods that work with continuous outputs (supports returned by the individual classifiers).

The former group consists of mainly voting algorithms [2], where majority voting is still the most popular method used so far. Other works in this area suggest to train the weights for controlling the level of importance assigned to each vote.

The latter group of combination methods is based on discriminants, or support functions. In general the support function is a measure of support given in favor of a distinguished class, as neural network output, *posterior* probability or fuzzy membership function. There are many approaches dealing with this problem as [7], in which the optimal projective fuser was presented, or [8] employing a probabilistic approach. Several analytical properties of aggregating methods were discussed e.g. in [9]. Basically, the aggregating methods, which do not require a learning procedure, use simple operators as the maximum, minimum, sum, product, or average value. Other works suggest to use a trained combiner in order to efficiently establish weights [6]. However although this is an efficient method, such an approach requires an extensive computational time and additional training dataset - both of which are not often available in real-life applications.

In this work, we introduce a novel method for simultaneous ensemble pruning and weighted combination. We propose novel weighted aggregation operators which do not require learning and have embedded pruning procedure that do not require any criterion to work. We work on modification of two popular operators: average of supports and maximum of supports. Their main drawback lies in lack of robustness to weak and irrelevant classifiers, and in minimizing the influence of other ensemble members. By using a Gaussian function to estimate the weights for the entire ensemble, we achieve a smooth method for reducing, but not eliminating the influence of weaker classifiers. At the same time by adjusting a threshold on the value of weights, we are able to prune the ensemble by discarding incompetent learners.

2 Classifier Combination Methods

As in this work we concentrate on weighted combination of continuous outputs, therefore let us assume that each individual classifier makes a decision on the basis of the values of support functions.

2.1 Weighted Aggregation

Let $\Pi = \left\{\Psi^{(1)}, \Psi^{(2)}, ..., \Psi^{(n)}\right\}$ be the pool of n individual classifiers and $F_{i,k}(x)$ stands for a support function that is assigned to class i ($i \in \mathcal{M} = \{1, ..., M\}$) for a given observation x and which is used by the classifier $\Psi^{(k)}$ from the pool Π.

The combined classifier $\Psi(x)$ uses the following decision rule

$$\Psi(x) = i \quad if \quad F_i(x) = \max_{k \in M} F_k(x), \tag{1}$$

where $F_k(x)$ is the weighted combination of the support functions of the individual classifiers from Π for the class k .

In this work, we assume that weights are dependent on classifier and class number. Weight $w_{i,k}$ is assigned to the k-th classifier and the i-th class. For a given classifier, weights assigned for different classes could be different. In our previous works, we have shown that this approach leads to a significant improvement over traditional methods [6]. With this, we can formulate our combination scheme as follows:

$$F_i(x) = \sum_{k=1}^{n} w_{i,k} F_{i,k}(x) \text{ and } \forall i \in \mathcal{M} \sum_{k=1}^{n} w_{i,k} = 1. \tag{2}$$

3 Untrained Ensemble Pruning and Weighted Combination

In this work, we propose new untrained aggregation operators which could exploit the competencies of the individual classifiers. The simple operators as maximum or average usually behave reasonably well but their work could be spoil by very imprecise estimators of the support functions used by only a few classifiers from a pool. Therefore we propose the modifications of the mentioned above operators which take into consideration all available support functions returned by the individual classifiers from the pool, but the functions which have the similar values to maximum or average have the strongest impact in the final value of the common support function calculated by using eq. 2. Additionally, we should notice that there may be some irrelevant classifiers in the pool and that for a large pool of classifiers most of the weights will become very small (in order to satisfy the condition from eq. 2). To deal with this problem, we propose to embed an ensemble pruning algorithm to eliminate incompetent classifiers. Then we normalize the weights for a reduced number of learners, thus increasing their level of influence over the ensemble decision. We propose to implement the pruning threshold ϕ, in order to discard all classifiers with assigned weights $w_{i,k} \leq \phi$.

The proposed operators are called NP-AVG and NP-MAX and can be calculated according to the Alg. 1. The only difference is the calculation of the $\overline{F}_i(x)$. For NP-AVG it is calculated according to

$$\overline{F}_i(x) = \frac{\sum_{k=1}^{N} F_{i,k}}{N}, \tag{3}$$

and for NP-MAX using the following formulae

$$\overline{F}_i(x) = \max_{k \in \mathcal{M}} F_{i,k}. \tag{4}$$

Algorithm 1. General framework for ensemble pruning and weight calculation

```
Require: $\Pi$ - pool of $n$ elementary classifiers
    $F_{i,k}(x)$ - support function value for each class $i$ returned by each individual classifier $k$ from $\Pi$
    $\phi$ - pruning threshold
Ensure: $w_{i,k}(x)$ - weights assigned to each support function $F_{i,k}(x)$ which could be used in eq.2
for $i := 1$ to $M$ do
    $w := 0$
    Calculate $\overline{F}_i(x)$ according to eq. 3 for NP-AVG or according to eq. 4 for NP-MAX
    for $k := 1$ to $n$ do
        $w_{i,k}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(F_{i,k}(x) - \overline{F}_i(x))}{2\sigma^2}\right)$
        $w := w + w_{i,k}(x)$
    end for
    for $k := 1$ to $n$ do
        if $w_{i,k} \leq \phi$
        discard the $k$-th classifier
    end for
    return pruned pool of $p$ classifiers
    for $k := 1$ to $p$ do
        $w := \frac{w_{i,k}(x)}{w}$
    end for
end for
```

The only parameters of the proposed operators is σ which equivalent of standard deviation in normal distribution, and a pruning threshold ϕ.

4 Experimental Investigations

The aims of the experiment was to check the performance of the two proposed aggregation operators N-AVG and N-MAX and to compare them with several popular methods for aggregating classifiers.

4.1 Datasets

In total we chose 10 well known datasets from the UCI Repository [4]. For datasets with missing values (autos, cleveland and dermatology), instances without full set of features available were removed.

4.2 Set-up

As a base classifier, we have decided to use neural network (NN) - realized as a multi-layer perceptron, trained with back-propagation algorithm, with number of neurons depending on the considered dataset: in the input layer equal to the number of features, in the output layer equal to the number of classes and in the hidden layer equal to half of the sum of neurons in previously mentioned layers. Each model was initialized with random starting values and their training process was stopped prematurely after 200 iterations, in order to assure the initial diversity of the pool and that we are working on weak classifiers.

The pool of classifiers used for experiments was homogeneous and consisted of 30 neural networks.

As a reference methods we decided to use popular classifier combination algorithms: majority voting (MV), maximum of support (MAX), average of supports (AVG) and product (PRO).

For a pairwise comparison, we use a 5x2 combined CV F-test [1]. For assessing the ranks of classifiers over all examined benchmarks, we use a Friedman ranking test [3] and Shaffer post-hoc test [5]. For all statistical analysis, we use the significance level $\alpha = 0.05$.

4.3 Results

Firstly, we need to establish the level of influence of value of pruning threshold ϕ on the quality of the ensemble. A grid search was performed for $\phi \in [0; 0.5]$ with step $= 0.05$. The best parameter values according to the final accuracy and the avg. size of the ensemble after pruning are given in Table 1. If $\phi = 0$, then no pruning was applied. We use the established values of this parameter for further comparisons.

Results of the experiments, presented according to the accuracy and reduction rate of the examined methods, are given in Table 2. Outputs of Shaffer post-hoc test over accuracy are given in Table 3.

Table 1. Selecting the value of pruning threshold ϕ, and is influence on the size of the ensemble. Numbers in brackets stands for a standard deviation in the ensemble size.

Dataset	Best ϕ value		Avg. size of the ensemble	
	NP-AVG	NP-MAX	NP-AVG	NP-MAX
Autos	0.00	0.00	30 (0.00)	30 (0.00)
Car	0.3	0.25	21 (2.45)	19 (3.03)
Cleveland	0.00	0.00	30 (0.00)	30 (0.00)
Dermatology	0.15	0.10	19 (3.23)	16 (2.09)
Ecoli	0.10	0.15	17 (4.23)	18 (2.78)
Flare	0.2	0.15	13 (1.28)	12 (2.03)
Lymphography	0.00	0.00	30 (0.00)	30 (0.00)
Segment	0.2	0.15	20 (4.02)	18 (3.11)
Vehicle	0.05	0.05	17 (2.26)	17 (1.84)
Yeast	0.15	0.05	12 (3.72)	11 (2.39)

Table 2. Comparison of the classifier combination methods, with respect to their accuracy [%]. Small numbers under accuracies stand for indexes of methods, from which the considered one is statistically superior. Last row stands for the avg. rank after the Friedman test.

Dataset	MV[1]	MAX[2]	AVG[3]	PRO[4]	NP-AVG[5]	NP-MAX[6]
Autos	62.34 –	65.84 1,3,4	64.23 1,4	63.05 –	67.54 *ALL*	66.32 1,3,4
Car	89.12 4,5	89.23 3,4,5	88.43 4,5	85.31 –	87.74 4	91.03 *ALL*
Cleveland	52.38 –	57.23 1,5,7	57.43 1,5,7	55.64 1	55.02 1	57.14 1,5,7
Dermatology	93.23 –	95.75 1,5,7	95.05 1,5	92.87 –	94.67 1	95.83 1,5,7
Ecoli	71.02 –	77.43 1,3,4,5	75.36 1,4,5	71.61 –	79.62 *ALL*	77.60 1,3,4,5
Flare	74.31 2,4,5	72.69 –	75.72 1,2,4,5,6,7	73.12 2	73.90 2,4,5	77.12 *ALL*
Lymphography	82.27 *ALL*	80.32 5	80.87 5	79.32 –	81.12 2,5	80.32 5
Segment	86.23 4,5	86.74 4,5	87.54 1,2,4,5,7	85.62 4	86.89 4,5	91.21 *ALL*
Vehicle	66.43 –	74.03 1,3,4,5,7	72.63 1,4,5,7	67.90 1	70.12 1,4,5	73.87 1,3,4,5,7
Yeast	43.41 –	52.36 1,3,4,5,7	49.78 1,4,5	45.02 1	50.11 1,4,5	57.98 *ALL*
Avg. rank	4.51	3.21	5.72	6.48	7.62	2.78

Let's present the conclusions derived from the experiments. The proposed operators behaved reasonably well and outperformed, with statistical significance, all of the traditional methods for 5 out 10 data sets. Modifications of the average operator N-AVG was significantly better than the original one in 3 out 10 experiments, while N-MAX (and N-AVG as well) was not significantly better than the original maximum operator. The Shaffer test confirmed that the combination rule which takes into consideration additional information (coming e.g. individual classifier accuracy) can outperform untrained operators. This confirmed our intuition, because the trained combination rule usually behave better than untrained one, what was confirmed in the literature. This test also showed that N-MAX is a slightly better than N-AVG, and what is interesting it can outperform most of the traditional untrained approaches except maximum operator. Analyzing characteristics of the used data benchmark sets we can suppose that proposed operators work well especially for the classification task where the number of possible classes is a quite high, but additional computer experiments should be carried out to confirm this dependency. Each of the proposed operators outperform majority voting for almost all data sets. We can conclude, that in the case of an absence of additional learning examples (which can be used to train the combination rule) the untrained aggregation is a better choice than voting methods. This observation is also known and confirmed by other researches as [11]. Our proposed methods allow to establish efficient weighted combination rules with a low computational complexity. Trained fusers require an additional processing time, which increases the complexity of the ensemble. Our methods, due to their low complexity, seem as an attractive proposition for real-life problems with limitations on processing time, e.g., ensembles for data streams.

Table 3. Shaffer test for comparison between the proposed combination methods and reference fusers. Symbol '=' stands for classifiers without significant differences, '+' for situation in which the method on the left is superior and '-' vice versa.

hypothesis	*p*-value
NP-AVG vs MV	+ (0.0423)
NP-AVG vs MAX	= (0.3895)
NP-AVG vs AVG	= (0.4263)
NP-AVG vs PRO	+ (0.0136)
NP-MAX vs MV	+ (0.0262)
NP-MAX vs MAX	= (0.4211)
NP-MAX vs AVG	+ (0.0249)
NP-MAX vs PRO	+ (0.0097)
NP-AVG vs NP-MAX	- (0.0314)

5 Conclusions

The paper presented two novel untrained aggregation operators which could be used in the case of the absence of additional learning material to train the combination rule. Otherwise the trained combination rule should be advised. The proposed methods could be valuable alternatives for the traditional aggregating operators which do not required learning and should be used in the mentioned above case instead of voting methods, of course in the case that we can access to the support function values of individual classifiers. The computer experiments confirmed that performances of the proposed methods are satisfactory compared to the traditionally untrained operators, especially for tasks when the number of possible classes is high. Therefore, we are going to continue the work on the proposed models, especially we would like to carried out the wider range of computer experiments which would define precisely the type of the classification tasks when the N-AVG and N-MAX could be used.

Acknowledgments. This work was supported by the Polish National Science Center under the grant no. DEC-2013/09/B/ST6/02264.

References

1. Alpaydin, E.: Combined 5 x 2 cv f test for comparing supervised classification learning algorithms. Neural Computation **11**(8), 1885–1892 (1999)
2. Biggio, B., Fumera, G., Roli, F.: Bayesian Analysis of Linear Combiners. In: Haindl, M., Kittler, J., Roli, F. (eds.) MCS 2007. LNCS, vol. 4472, pp. 292–301. Springer, Heidelberg (2007)
3. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
4. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml
5. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Inf. Sci. **180**(10), 2044–2064 (2010)
6. Jackowski, K., Krawczyk, B., Woźniak, M.: Improved adaptive splitting and selection: the hybrid training method of a classifier based on a feature space partitioning. Int. J. Neural Syst. 24(3) (2014).
7. Rao, N.S.V.: A Generic Sensor Fusion Problem: Classification and Function Estimation. In: Roli, F., Kittler, J., Windeatt, T. (eds.) MCS 2004. LNCS, vol. 3077, pp. 16–30. Springer, Heidelberg (2004)
8. Rokach, L., Maimon, O.: Feature set decomposition for decision trees. Intell. Data Anal. **9**(2), 131–158 (2005)
9. Wozniak, M.: Experiments on linear combiners. In: Pietka, E., Kawa, J. (eds.) Information Technologies in Biomedicine. AISC, vol. 47, pp. 445–452. Springer, Berlin / Heidelberg (2008)
10. Woźniak, M., Graña, M., Corchado, E.: A survey of multiple classifier systems as hybrid systems. Information Fusion **16**, 3–17 (2014)
11. Wozniak, M., Zmyslony, M.: Designing combining classifier with trained fuser - analytical and experimental evaluation. Neural Network World **20**(7), 925–934 (2010)