Ensemble Based on Data Envelopment Analysis

So Young Sohn & Hong Choi

Department of Computer Science & Industrial Systems Engineering,
Yonsei University, Seoul, Korea
Tel) 82-2-2123-4014, Fax) 82-2- 364-7807
E-mail: sohns@yonsei.ac.kr

Abstract. There has been much research to evaluate the efficiency of various data fusion/ensemble approaches. However, when combining individual classifiers for fusion or ensemble purposes, typically only misclassification rate has been considered as a performance measure. This might be risky especially when the class distribution is skewed or when the costs associated with both Type I and II errors are significantly different from each other. For this kind of situation, consideration of additional performance measures such as sensitivity, specificity, false negative or positive errors are needed. In this paper, we propose to use DEA in order to find the weights involved in multi-attributes performances of each classifier as an element of a data ensemble algorithm. This algorithm is expected to serve general purposes of classification.

1. Introduction

Data mining is the process of extracting valid, previously unknown, and ultimately comprehensible information from large databases and using it to make crucial business decisions. Effective mining necessary information and knowledge from a large database has been recognized as a key research topic by many practitioners in the field of data-based marketing. Algorithms often used for data mining can be classified into one of the following areas: artificial neural network, machine learning, and classical statistical models. It has been reported that the classification accuracy of the individual algorithm can be improved by combining the results of several classifiers. Data fusion techniques try to combine classification results obtained from several single classifiers and are known to improve the classification accuracy when some results of relatively uncorrelated classifiers are combined. Data ensemble combines various results obtained from a single classifier fitted repeatedly based on several bootstrap resamples. The resulting performance is known to be more stable than that of a single classifier.

There has been much research to evaluate the efficiency of various data fusion/ensemble approaches. However, when combining individual classifiers for fusion or ensemble purposes, typically only one attribute, that is misclassification rate, has been considered as a performance measure. This might be risky especially when the class distribution is skewed or when the costs associated with both Type I and II errors are significantly different from each other. For this kind of situation, consideration of additional performance measures such as sensitivity, specificity, false negative or positive errors are needed. Then it becomes a multi-attribute decision making problem in terms of finding class information for a given case based on different weights on various performance measures. Subsequent question raised is how to find such weights.

In this paper, we propose to use DEA in order to find the weights involved in multi-attributes performances of each classifier as an element of a data ensemble algorithm.

Data Envelopment Analysis (DEA) has been frequently applied to assess the efficiency of several decision making units (DMU) which have multiple inputs as well as outputs. By way of DEA, one can find the efficiency score of each DMU and can figure out the set of efficient DMUs based on the set of non-dominated solution. In addition, DEA provides inefficient DMUs with the benchmarking point and has advantages over alternative parametric approaches such as regression or ratio analysis [2].

Organization of this paper is as follows. In section 2, we briefly summarize the established data ensemble techniques along with related literature. In section 3, we introduce the ensemble based on DEA. In section 4, we illustrate an example of the proposed method using the road traffic accident data. In section 5 we summarize our findings.

2. Literature Review

We first review data ensemble literature. Ensemble algorithms can be divided into two types: those that adaptively change the distribution of the bootstrap training set based on the performance of previous classifiers, as in Boosting methods or Arcing (Adaptive resampling and Combining) and those that do not, as in Bagging (Bootstrap AGGregatING)).

Bagging (Bootstrap AGGregatING) algorithm introduced by Breiman [1] votes classifiers generated by different bootstrap samples. A bootstrap sample is generated by uniformly sampling N instances from the training set with replacement. Detailed procedure is as follows:

Step 1. Suppose $f(x,t)$ is a classifier, producing an M-vector output with 1(one) and M-1 (zero), at the input point *x*.

Step 2. To bag $f(x,t)$, we draw bootstrap samples $T_m = (t_{1m}, t_{2m}, \cdots, t_{Nm})$ each of size N with replacement from the training data.

Step 3. Classify input point *x* to the class *k* with largest "vote" in $f^k_{bagging}(x,t)$ as follows.

$$f^k_{bagging}(x,t) = \frac{1}{M}\sum_{m=1}^{M} f^k_m(x,t) \quad (1)$$

The basic idea of Bagging is to reduce the deviation of several classifiers by voting the classified results due to bootstrap resamples.

Arcing (Adaptive Resampling and Combining) is designed exclusively for classification problem, which is developed by Freund and Shapire [3] in the name of "boosting" but Breiman renamed it as "arcing". Basic idea of it is almost like bagging which tries to reduce the deviation of several classifiers by voting the classified results due to bootstrap resamples. Arcing takes unequal probability bootstrap samples, that is the probability of a training example being sampled is not uniform, but depends on the training error of previous predictors. In General, the classification procedure of arcing (Adaboost M1) can be summarized as follows:

Step 1. Sample with replacement from T with probabilities $P_m(i)$ (where $P_m(i)=1/N$ and $i=1,2,\cdots,N$) (m=1,…, M) and construct the classifier f_m using the resampled set T_m of size N.

Step 2. Classify T_m using f_m and let d(i)=1 if example i is incorrectly classified, else d(i)=0.

Step 3. Calculate the ε_m and β_m as follows.

$$\varepsilon_m = \sum_{i=1}^{N} P(i)d(i), \quad \beta_m = \frac{(1-\varepsilon_m)}{\varepsilon_m} \tag{2}$$

Step 4. Update probabilities $P_{m+1}(i)$ by using the following formula.

$$P_{m+1}(i) = \frac{P_m(i)\beta_m^{d(i)}}{\sum P_m(i)\beta_m^{d(i)}} \tag{3}$$

Step 5. Let m=m+1 and go to Step 1 if m<M.

Step 6. Take a weighted vote of the classifications, with weights log (β_m).

Quinlan [7] reported results of applying both bagging and boosting by decision trees (C4.5) on 18 data sets. Although boosting generally increases accuracy more than bagging, it also produces severe degradation on some data sets. The authors' further experiment showed that such deterioration in general performance of boosting is resulted from over-fitting a large number of trials that allow the composite classifier to become very complex. Instead of using the fixed weight for the vote of classifier, they suggested using the voting weight of each classifier to vary in response to the confidence with which the instance is classified. Trials over a diverse collection of datasets under their suggestion reduced the downside of classification accuracy and also led to slightly better results on most of the datasets considered.

Opitz and Maclin [6] presented an empirical evaluation of Bagging and Boosting as methods for creating an ensemble of neural networks and decision-tree with 14 data sets. The authors found out that Bagging is appropriate for most problems, but when properly applied, Boosting may produce even larger gains in accuracy. Their results also showed that the advantages and disadvantages of both Bagging and Boosting depend only on the domain to which they are applied, instead of the type of classifier.

Hansen [4] compared five meta machine learning methods which employ neural networks as an ensemble member: three from ensemble methods (Simple, Bagging and Adaboost) and two from mixture expert methods (XuME and Dynco). The empirical results showed that the cooperative error function of Dynco is superior to the competitive error function of the others.

Kohavi and Wolpert [5] proposed bias and variance decomposition for misclassification error, when there are only two levels of class. The authors showed how estimating the terms in the decomposition using frequency counts leads to biased estimates and explained how to get unbiased estimators to overcome such major shortcomings as obtaining potentially negative variance. They then gave some examples of the bias-variance tradeoff using two machine learning algorithms applied to data available in several UC-Irvine repository.

3. Ensemble Based on DEA

When multiple classifiers are obtained, we suggest to combine those results using the weight reflecting multi-attribute performance measures such as sensitivity, specificity, bias and variance of misclassification rate defined as follows:

$$Sensitivity = \frac{(Number\ of\ observations\ that\ predict\ the\ event\ 1\ correctly)}{(Number\ of\ observations\ that\ represent\ the\ event\ 1)} \tag{4}$$

$$Specificity = \frac{(Number\ of\ observations\ that\ predict\ the\ event\ 0\ correctly)}{(Number\ of\ observations\ that\ represent\ the\ event\ 0)} \tag{5}$$

$$bias^2 \equiv \frac{1}{2}\sum_{y \in Y}\left[P(Y_F = y|x) - P(Y_H = y|x)\right]^2 \tag{6}$$

$$\mathrm{var}\,iance_x \equiv \frac{1}{2}\left(1 - \sum_{y \in Y} P(Y_H = y|x)^2\right) \tag{7}$$

where $P(Y_F = y|x)$ is the probability that the outcome of a given case with input x is y while $P(Y_H = y|x)$ is the probability that the outcome of a given case with input x is classified as y.

As a means to obtain an individual weight for each performance measure, DEA is proposed. Data Envelopment Analysis (DEA) has been frequently applied to assess the efficiency of several decision making units (DMU) which have multiple inputs as well as outputs. In our case, individual classifier is considered as DMU while we consider outputs as the four performance measures of each classifier along with constant inputs. By way of DEA, one can find the efficiency score of each DMU and can figure out the set of efficient DMUs based on the set of non-dominated solution.

In addition, DEA provides inefficient DMUs along with benchmarking points and has advantages over alternative parametric approaches such as regression or ratio analysis [2]. We suggest DEA ensemble as follows:

Step 1. Choose the machine learning algorithm to be used as a classifier.

Step 2. Generate training set T_m by sampling from T with replacement, where the probability for sampling training case $\boldsymbol{i}$ is $P_m(i)$.

Step 3. Construct the classifier $f_m(x,t)$ using the resampled set T_m of size N.

Step 4. Evaluate the classifier $f_m(x,t)$ in terms of multi-attributes y_{rm} where it represents the *r*th output variable of *m*th classifier. In this study, we consider four attributes–sensitivity y_{rm} (y1), specificity (y2), the squared bias of misclassification (y3) and the variance of misclassification (y4).

Then for each unit "*o*" we want to find the best weight v_i that maximize the weighted output by solving the following mathematical programming model:

$$\begin{aligned} \max\ & h_o = \sum_{r=1}^{s} v_r y_{rm} \\ s.t\ & \sum_{r=1}^{s} v_r y_{rm} \le 1 \quad for\ m = 1,\cdots,M \\ & v_r > 0, \quad s = 1,2,3,4 \end{aligned} \tag{8}$$

Each DMU_o is assigned the highest possible efficiency score that the constraints allow from the available data by choosing the appropriate virtual multipliers (weights) for the outputs. Let h_o* denote the optimal value of h_o where $0 \le h_o^* \le 1$. One can say that $h_o^*=1$ and the complementary slackness conditions of linear programming are met if and only if unit *o* is efficient relative to other units considered. On the other hand, if $h_o^*<1$, then this unit is considered as inefficient which could not achieve a higher rating relative to the reference set to which it is being compared.

Step 5. Once a set of the efficiency scores of M classifiers $h_m = (h_1,\cdots,h_M)$ is found, normalize it to $\hat{h} = \left(\hat{h}_1, \hat{h}_2, \cdots, \hat{h}_M\right)$ so that $\sum_{m=1}^{M} \hat{h}_m = 1$ and take a weighted vote of the classifications, with weights $\hat{h}_m$ as follows.

$$f_{bagging}(x,t) = \sum_{m=1}^{M} \hat{h}_m \cdot f_m(x,t) \tag{9}$$

4. Numerical Example

In this section, we apply the proposed DEA ensemble algorithm to the actual data for illustration. Sohn and Shin [9] used individual algorithms such as neural network and decision tree to classify the severity of road accidents occurred in Seoul, Korea in 1996. Input variables used for classification of two levels of severity (bodily injury and property damage) are road width, shape of car-body, accident category, speed before the accident, violent drive, and protective device. Detailed levels of these input variables are displayed in Table 1. These variables were selected using decision tree and all turned out to have better explanatory power than variables representing weather conditions. A sample of 11564 accidents was taken and 60% of them were used for training while rest of them was used for validation, respectively. Correct classification rates obtained by both classification models were not significantly different. In order to increase the classification performance, we use DEA ensemble introduced in the previous section.

Table 1 Input Variable Description

<table>
<tr><th colspan="3">Accident Classification</th><th colspan="2">Road Width</th><th colspan="3">Car Shape</th><th colspan="2">Accident Type</th><th colspan="2">Velocity</th></tr>
<tr><td rowspan="4">Injury Accident</td><td>01</td><td>Death</td><td>01</td><td>Service Area</td><td rowspan="4">Passenger Car</td><td>01</td><td>Bus</td><td rowspan="4">01~06</td><td rowspan="4">Man to Car</td><td>01</td><td>below 10km</td></tr>
<tr><td>02</td><td>Major Injury</td><td>02</td><td>below 3m</td><td>02</td><td>Microbus</td><td>02</td><td>below 20km</td></tr>
<tr><td>03</td><td>Minor Injury</td><td>03</td><td>over 3m</td><td>03</td><td>1BOX</td><td>03</td><td>below 30km</td></tr>
<tr><td>04</td><td>Injury Report</td><td>04</td><td>over 6m</td><td>04</td><td>A Sedan</td><td>04</td><td>below 40km</td></tr>
<tr><td>No Injury Accident</td><td>05</td><td>Property Damage</td><td>05</td><td>over9m</td><td rowspan="10">Cargo Car</td><td>11</td><td>Ice Truck</td><td rowspan="5">21~27</td><td rowspan="5">Car to Car</td><td>05</td><td>below 50km</td></tr>
<tr><td></td><td></td><td></td><td>06</td><td>over 13m</td><td>12</td><td>Dump Truck</td><td>06</td><td>below 60km</td></tr>
<tr><td></td><td></td><td></td><td>07</td><td>over 20m</td><td>13</td><td>Concrete Truck</td><td>07</td><td>below 70km</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>14</td><td>Tank</td><td>08</td><td>below 80km</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>15</td><td>Wash Car</td><td>09</td><td>below 90km</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>16</td><td>Special Purpose</td><td rowspan="6">31~42</td><td rowspan="6">Car Alone</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>17</td><td>Container</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>18</td><td>1BOX</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>19</td><td>Light Van</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>20</td><td>Truck Class</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td>00</td><td>Unknown</td><td></td><td></td></tr>
</table>

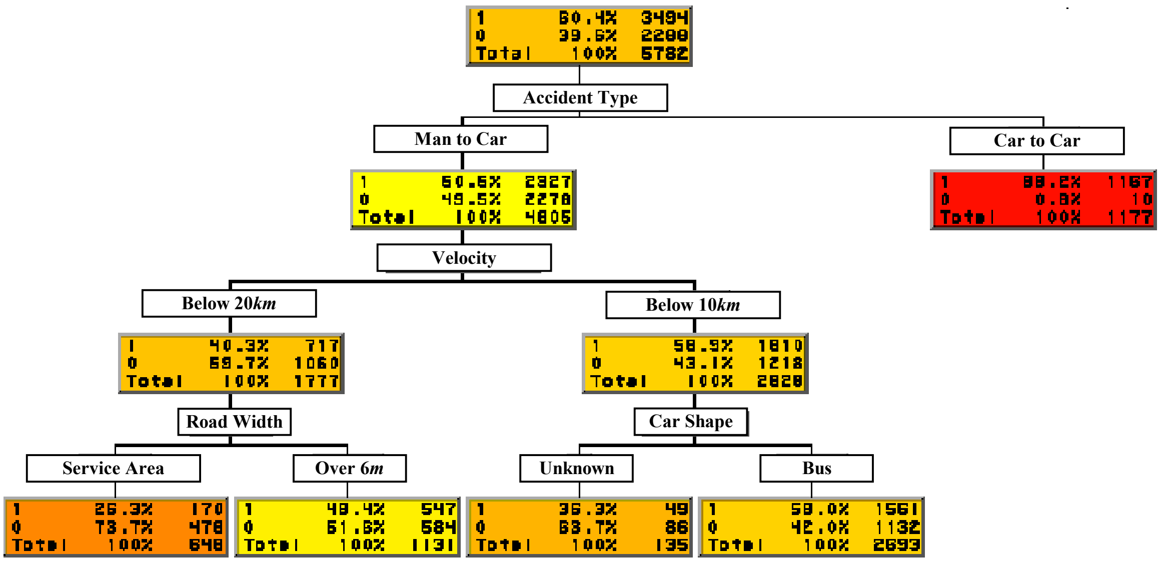

Figure 1 Decision Tree based on First Bootstrap Resampled Training Data Set

We generate 10 Bootstrap resamples and fit decision tree for each resample. Figure 1 shows a result of fitted tree. This fitted tree classifier is evaluated in terms of sensitivity, specificity, bias and variance of misclassification rate. Obtained values can be used for outputs for DEA. Summarized values for all 10 fitted tree classifiers are displayed in Table 2.

We use SAS/OR [8] to solve mathematical programming for DEA and efficiency score obtained for each classifier is summarized in Table 3. Apparently, classifiers 2,5,6,8 and 9 turn out to be efficient and accordingly their weights are higher than the rest of the remaining classifiers.

We then apply these weights to the results of individual classifiers to obtain DEA ensemble outcome. Finally, DEA ensemble is compared to a singletree result with respect to multiple performance measures.

Table 2 Performance Measures of Each Bootstrap Classifier

	Sensitivity	Specificity	Bias^2	Variance
classifier 1	0.91	0.32	0.0484	0.1476
classifier 2	0.89	0.38	0.0194	0.1716
classifier 3	0.82	0.50	0.0081	0.2139
classifier 4	0.88	0.39	0.0289	0.1771
classifier 5	0.83	0.46	0.0121	0.1030
classifier 6	0.82	0.49	0.0100	0.2100
classifier 7	0.74	0.63	0.0001	0.2419
classifier 8	0.84	0.46	0.0144	0.2016
classifier 9	0.80	0.51	0.0049	0.2211
classifier10	0.68	0.70	0.0064	0.2496

Table 3 Efficiency Score and Weight for Each Classifier

	Efficiency Score	Weight for Each Calssifier
classifier 1	0.999275	0.101170921
classifier 2	1.000000	0.101244323
classifier 3	0.998996	0.101142674
classifier 4	0.998044	0.101046289
classifier 5	1.000000	0.101244323
classifier 6	1.000000	0.101244323
classifier 7	0.991065	0.100340
classifier 8	1.000000	0.101244323
classifier 9	1.000000	0.101244323
classifier10	0.889717	0.090078795

DEA ensemble appears to be better in terms of sensitivity and variance. In our case, there was not much variation among the ten efficiency scores and therefore the weights were almost even. Note that when the weights are the same, DEA ensemble would be equivalent to Bagging.

Table 4 Comparisons of the Classification Results between DEA Ensemble and Decision Tree

	Sensitivity	Specificity	Bias^2	Variance
DEA Ensemble	0.90	0.35	0.0207	0.1485
Decision Tree	0.73	0.64	0.0011	0.2003

6. Conclusion

In this paper, we suggest DEA ensemble, which reflects more than one performance measures for voting. This is a generalized version of Bagging and is expected to have maximum utilization when there is much variation in the efficiency scores of individual classifiers. One drawback of the proposed DEA ensemble has, as the other ensemble is that the combined rule cannot be explicitly stated.

DEA ensemble concept suggested can be extended to Arcing by reflecting multiple performances when assigning new selection probabilities. This is left as further study areas.

References

[1] Breiman, L., "Bagging, Boosting, and C4.5", ftp://ftp.stat.berkeley.edu/pub/users /breiman, 1996.

[2] Charnes, A., Cooper, W. W., and Rhodes, E., Measuring the efficiency of decision making units, *European Journal of Operational Research*, 2(6):429-444, 1978.

[3] Freund, Y. and Shapire, R. E., A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences*, 55:119-139, 1995.

[4] Hansen, J. V., Combining predictors: comparison of five meta machine learning methods, ***Information Sciences***, 119:91-105, 1999.

[5] Kohavi, R., Wolpert, D. H., Bias plus variance decomposition for zero-one loss function, Proceedings of the Thirteenth International Conference on Machine Learning, 1996.

[6] Opitz, D. W. and Maclin, R. F., An empirical evaluation of bagging and boosting for artificial neural networks, International conference on neural networks, 3:1401-1405, 1997.

[7] Quinlan, J. R., Bagging, Boosting, and C4.5 In Proceedings of the Thirteenth National Conference on Artificial Intelligence, 1996. Available on line at: http://www.cse.unsw.edu.au/~quinlan/.

[8] SAS/OR Manual, SAS Institute, Cary, NC, USA, 1992.

[9] Sohn, S. Y. and Shin, H. W., Pattern Recognition for Road Traffic Accident Severity in Korea, *Ergonomics*, 44(1):107-117, 2001.