A DEA Approach for Model Combination

Zhiqiang Zheng
A.G. School of Management
University of California
18 Anderson Hall
Riverside, CA 92521
+1(909)787-3779
eric.zheng@ucr.edu

Balaji Padmanabhan
The Wharton School
University of Pennsylvania
3730 Walnut Street
Philadelphia, PA 19104
+1(215)573-9646
balaji@wharton.upenn.edu

Haoqiang Zheng
Department of Computer Science
Columbia University
1214 Amsterdam Ave
New York, NY 10027
+1(212)939-7059
hzheng@cs.columbia.edu

ABSTRACT

This paper proposes a novel Data Envelopment Analysis (DEA) based approach for model combination. We first prove that for the 2-class classification problems DEA models identify the same convex hull as the popular ROC analysis used for model combination. For general *k*-class classifiers, we then develop a DEA-based method to combine multiple classifiers. Experiments show that the method outperforms other benchmark methods and suggest that DEA can be a promising tool for model combination.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Applications – *Data Mining*; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Experimentation

Keywords

Model Combination, Data Envelopment Analysis, ROC

1. INTRODUCTION

Combining multiple models has attracted interest across a variety of fields including management science, statistics, data mining and machine learning [1, 4, 6, 9, 12]. Studies have shown that combined models consistently outperform individual models for classification tasks [4, 9]. However, most methods for model combination are still ad-hoc in nature. How individual models are related to each other and how they should contribute to combination remain active research questions.

Recently Provost and Fawcett [13] addressed the above problems for classification models by using ROC (Receiver Operating Characteristics) analysis.

KDD'04, August 22–25, 2004, Seattle, Washington, USA.

An ROC space is specified by the misclassification matrix along two dimensions – the TP (True Positive Rate) and FP (False Positive Rate) of the classifiers. Conversely, a particular classifier is represented by a pair of TP and FP in the ROC space. In [13] a set of classifiers is mapped into the ROC space specified by TP and FP. By combining the models lying on the convex hull formed by the individual classifiers in the ROC space it is shown that the combined classifiers yield a robust hybrid classifier that theoretically could outperform all individual classifiers in terms of the TP and FP tradeoff. However one aspect of their approach is that traditional ROC analysis can only deal with two-class (binary) classification problems.

It was pointed out that ROC analysis theoretically can be extended to *k*-dimension classification problems [14]. For a *k*-class problem, the misclassification cost matrix for a particular classifier represents a point in a (k^2-k) dimensional space. The set of potentially optimal classifiers forms a convex hull in this space. Optimal classifiers can then be located via linear optimization where the facets of the ROC convex hull (ROCCH) define the optimal constraints. Thus as [11] concluded, in principle, it should be possible to apply ROC type of analysis techniques to multi-class cases.

In this paper, we introduce a method from the Operations Research field called Data Envelopment Analysis (DEA) and show how it can be used to select and combine models. At its core, DEA addresses the key issue of determining the efficiencies of various producers or DMUs (Decision Making Units) in converting a set of inputs into a set of outputs [2, 8]. Each DMU corresponds to a data point, the attributes of which can be partitioned into inputs and outputs. DEA uses a linear programming approach to determine how efficient each DMU is in converting inputs into outputs. The points lying on the convex hull are termed as efficient DMUs. DEA deals with not only 2-class problems, but *k*-dimensional problems (multiple inputs and multiple outputs) as well. We show that for a 2-class problem, the convex hull identified by the ROC analysis is equivalent to the efficient frontier identified by DEA. More importantly, we show that DEA can nicely extend ROCCH to *k*-class problems.

The main contribution of this paper is that it presents a new method for model combination using DEA analyses. We propose a novel way of formulating a classification problem into a DEA formulation by identifying the inputs and outputs for DEA. This approach can be used to combine classifiers for the more general

k-class classification problems extending the ROCCH [13] approach. We show that model combination using DEA significantly outperforms conventional approaches for a variety of classification problems.

In the following section, we first review the relevant literature in model combination. In section 3 we prove the equivalence of the ROCCH and DEA approaches in dealing with 2-class classification problems. We develop a DEA-based method to combine *k*-class classifiers in section 4. Experiments and results are presented in section 5.

2. LITERATURE REVIEW

The problem of combining multiple models has been addressed across a variety of fields including management science, statistics, forecasting and machine learning. For the lack of space, we only describe the ROC analysis [13] and the DEA approach [2, 8]. Readers are referred to [1, 4, 6, 9, 12] for further review.

2.1 ROC Analysis

ROC (Receiver-operating characteristics) analysis deals with two-class classification problems. One class is normally referred to *positive* and the other *negative*. A particular classification model (i.e. classifier) maps instances to predicted classes. The predictions could either be *true* or *false*. Hence *true positive rate* (TP) and *false positive rate* (FP) are defined as:

TP = positives correctly classified/ total positives
FP = negatives incorrectly classified/ total negatives

An ROC space thus is two-dimensional, normally specified by FP as *x*-axis and TP as *y*-axis. Each data point in this space represents a tradeoff of between TP and FP yielded from a particular model. The convex hull (frontier) of this ROC space forms a curve that represents the most efficient TP and FP tradeoff. Models lying on the frontier are the best since they convert FP into TP in the most efficient fashion. The models inside the convex hull are dominated by the models on the convex hull. In [13] an algorithm, ROC Convex Hull (ROCCH), is developed to identify the best models on the frontier and then [13] used the convex hull to construct a hybrid classifier. Users first make a cost assumption (e.g. TP is 4 times more important than FP) with respect to TP and FP. A hybrid classifier consists of the set of points where the iso-slope implied by the cost matrix is tangential to the ROC hull. It is also shown in [13] that under any target cost and class distributions, a robust classifier will perform at least as well as the best classifier.

2.2 Data Envelopment Analysis (DEA)

Here we briefly discuss DEA, the problem it was developed to address and the basic model it uses. DEA is a linear programming based approach that constructs an efficient frontier (envelopment) over the data, and calculates each data point's efficiency relative to this frontier [8]. DEA assumes that the variables of the data can be logically divided into inputs and outputs. Each data point corresponds to a decision making unit (DMU), or a producer in practice. The "decision" of a unit is to convert inputs into outputs as efficiently as possible.

DEA uses a linear programming approach to identify the efficient DMUs, those units that make the most efficient use of inputs to produce outputs. The efficient units consist of a frontier among all DMUs. The efficiencies of the DMUs are measured by projecting to this frontier.

The DEA model in its original form represented the performance or efficiency of the DMU as the ratio of weighted outputs to weighted inputs [8]. To date, the DEA literature has developed numerous models and detailed discussions can be found in [7, 8]. Essentially, various models for DEA seek to establish which subset of DMUs determines an envelopment surface and address how to characterize each DMU by an efficiency score. One of the classic model formulations, the BCC input-oriented model [2], is presented below.

Suppose N data points (DMUs) are to be evaluated. Each data point consists of K inputs and M outputs. The $K \times N$ input matrix, X, and the $M \times N$ output matrix, Y, represent the data. Note that in this formulation, the columns represent the data points and the rows are the variables (these are inputs in the case of X and are outputs in the case of Y). Hence for the i^{th} DMU the inputs are represented by the $K \times 1$ vector x_i and the outputs by the $M \times 1$ vector y_i. A good account of the BCC model can be found in [2], we directly present the formulation below:

$$\min_{\lambda,\theta} \theta$$
$$\text{such that} \quad -y_i + Y\lambda \geq 0$$
$$\theta x_i - X\lambda \geq 0$$
$$1'\lambda = 1$$
$$\lambda > 0 \qquad (1)$$

Where $1'$ is an $1 \times N$ vector of ones, θ is a scalar and λ is an $N \times 1$ vector of constants. In the BCC model formation, θ and λ are the only two variables to be optimized. All the other values (X, Y) are given by the data. A linear program for optimizing this essentially identifies a convex hull for all the data points and computes an efficiency score θ for each individual data point. The value of θ is bounded between 0 and 1, with 1 indicating a point on the frontier. Note that the linear programming problem must be solved N times, once for each DMU in the sample.

DEA has been successively applied to a variety of applications such as studying the efficiency of commercial banks, to anticipate the consequences of school reforms and to investigate online customers shopping efficiency [8]. This paper, however, is the first attempt to apply DEA to study model combination.

3. DEA AND ROC CONVEX HULLS

Both ROCCH and DEA identify a convex hull. In this section we show that the two convex hulls indeed are identical for a 2-class classification problem.

3.1 ROC Convex Hull Algorithm

ROC Convex Hull [13] has several distinct characteristics:

1. It is two dimensional, characterized by the TP and FP of a classifier.

2. It goes through (0,0) and (1,1)[1], and both TP and FP values are bounded within (0,1).
3. It identifies the convex hull in the northwest area, i.e. above the line TP = FP.
4. It is monotonic and increasing.

Two-dimensional convex hull algorithms, are essentially variants of the "triangle test" - for a data point to be on the convex hull, it must *not* be within the triangle formed by any three data points [3]. But the computation complexity for this test is $O(N^4)$ for N records. ROCCH uses an algorithm similar to the Quickhull algorithm [3, 13], the complexity of which reduces to $O(N \log N)$. The property of monotonicity further simplifies the core of the algorithm as follows:

> *Assume (X_j, Y_j) and (X_{j+1}, Y_{j+1}) are two adjacent points on the current identified convex hull. To evaluate a particular data point (x_i, y_i) where $X_j \le x_i \le X_{j+1}$, we only need to check if (x_i, y_i) is below or above the line formed by (X_j, Y_j) and (X_{j+1}, Y_{j+1}). If (x_i, y_i) is below the line, maintain the old convex hull and check new data points. Otherwise, modify the old convex hull to include this new point*[2].

3.2 DEA Convex Hull

It is not immediately clear how to represent an ROC problem by a DEA model. Here we take the approach that a classifier can be considered to be a DMU that makes tradeoffs between TP and FP. A classifier can be perceived as a decision maker that attempts to maximize the output, TP, at the expense of the input, FP. Thus the ROC problem can be formulated into a DEA problem as follows: a 2-class classifier is a DMU in DEA that takes FP as inputs and TP as outputs.

We choose the classic BCC formulation [2] as the DEA model. As mentioned before note that the formulation below has to be solved for each classifier separately to get its efficiency score. Among a set of N classifiers for evaluating a particular classifier i (with FP=x_i and TP=y_i) the one-input one-output BCC model simplifies as:

$$\min_{\lambda,\theta} \theta$$

$$\text{such that} \quad \sum_{j=1}^{N} \lambda_j y_j \ge y_i$$

$$\sum_{j=1}^{N} \lambda_j x_j \le \theta x_i$$

$$\sum_{j=1}^{N} \lambda_j = 1$$

$$\lambda_j > 0 \qquad (2)$$

According to [8], the i^{th} DMU is to be on the efficiency frontier, if and only if the solution θ is equal to 1. Thus in order to test whether ROCCH identifies the same convex hull, we just need to prove one of the two cases:

1. A model is on the convex hull from ROCCH if and only if it has the property $\theta = 1$ in the DEA model.
2. A model is *not* on the convex hull from ROCCH if and only if it has the property $\theta < 1$ in the DEA model.

We will show that the second holds.

3.3 DEA and ROC Convex Hulls are Identical

We prove in this section that DEA identifies the same convex hulls as ROCCH, for a 2-class classification problem. We do this by using the second test: an inefficient point in the ROC space has efficiency score $\theta < 1$ in DEA.

The best way to show this is to prove it graphically (see Figure 3.1). Suppose a convex hull has been identified by ROCCH. Without loss of generality, assume (X_j, Y_j) and (X_{j+1}, Y_{j+1}) are two points on the curve. And (x_i, y_i) is an inefficient point, $X_j < x_i < X_{j+1}$, that is below the line in Figure 3.1 (FP as x-axis and TP as y-axis.) Now the test simplifies as: is the value θ of point (x_i, y_i) less than one in a DEA model?

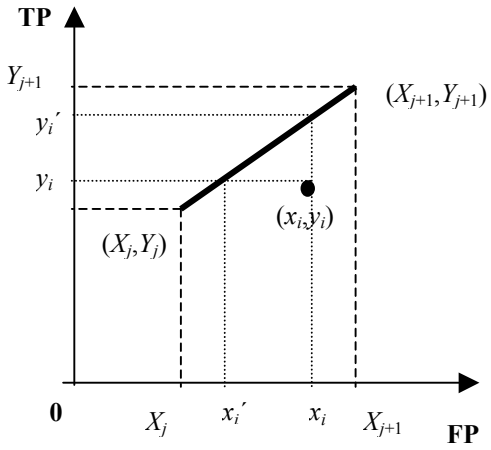

Figure 3.1: An inefficient point in the ROC space

It is easy to see that this is indeed the case. Clearly point (x_i, y_i) satisfies the first condition of the BCC model (see formulation (2)), $\Sigma \lambda_j Y_j \ge y_i$, for (x_i, y_i) is below the line. Input-oriented BCC model projects a point to the left. Thus (x_i', y_i) is the projection of (x_i, y_i) on the convex hull and $x_i' < x_i$ (see Figure 3.1). According to the property of linear combination, x_i' must be equal to $\Sigma \lambda_j X_j$, i.e., $x_i' = \Sigma \lambda_j X_j$. Thus $\Sigma \lambda_j X_j < x_i$ or equivalently $\theta < 1$.

The converse of the statement also holds, establishing the *iff* assertion: i.e. a DMU with $\theta < 1$ is an inefficient point in the ROC space. Suppose a classifier (x_i, y_i) has $\theta < 1$ in the DEA formulation, then $\Sigma \lambda_j X_j < x_i$ and $\Sigma \lambda_j Y_j \ge y_i$. This implies that the point (x_i, y_i) is below the two adjacent classifiers (X_j, Y_j) and $(X_{j+1}$

[1] i.e. when a naïve model classifies all instances as negative or positive respectively.

[2] A literal summarization of the algorithm as described in the perl script for ROCCH [13], available at http://www.hpl.hp.com/personal/Tom_Fawcett/ROCCH.

,Y_{j+1}) on the convex hull, where $X_j < x_i < X_{j+1}$ (see Figure 3.1). Based on the equivalence of the two tests described in 3.2, the following two propositions are also true: 1) A point on the convex hull of the ROC space has efficiency score $\theta = 1$ in DEA and 2) A DMU with $\theta = 1$ is an efficient point in the ROC space.

In summary, for 2-class problems, the convex hull identified by ROCCH is identical to the one identified by a DEA model.

4. A DEA APPROACH TO COMBINING *K*-CLASS CLASSIFIERS

An important observation that will significantly help address model combination for multi-class problems is that DEA is not limited to 2-dimension problems. DEA's ability to deal with many inputs and many outputs suggests that DEA could extend ROCCH to *k*-class problems. In this section, we develop a DEA approach to combine classifiers for this general case. The method extends ROC analysis to multi-class classification tasks.

The first step of a typical DEA application is to identify the inputs and outputs. As discussed for the 2-class problem, we model a classifier as a DMU that attempts to produce TP at the expense of FP. However, it is not as clear what makes inputs and outputs for a general *k*-class classification problem.

In [11], it is pointed out that for a *k*-class problem, a particular classifier could be represented by a k^2 confusion matrix (or misclassification cost matrix). Let *C* denote the $k \times k$ confusion matrix. Each element c_{ij} represents the frequency that class *i* is misclassified as *j*, $i \neq j$ and $i, j \in (1, k)$; and when $i = j$, c_{ij} is the frequency that class *i* is correctly classified. The frequency of a particular cell can be represented either by the actual number of misclassified instances or by a row percentage of them.

Hence, a classifier can be perceived as a DMU that attempts to maximize c_{ii} and minimize c_{ij} , $i, j \in (1, k)$. In DEA terms, a classifier can be viewed as a decision maker that attempts to produce diagonal elements c_{ii} at the expense of off-diagonal elements c_{ij}. As such, c_{ij} can be interpreted as the inputs and c_{ii} as the outputs in a DEA model. Notice that the actual degree-of-freedom of *C* is k^2-*k* since the values of each row in the matrix add up to a fixed value. This is evident if we take off the *K* diagonal elements. In our experiments, we actually remove the K^{th} column since we need to use diagonal elements as outputs in our DEA formulation. Thus, a *k*-class classifier $\{f(x, y), y \in (1, k)\}$ is a DMU in DEA that takes c_{ij} as inputs and c_{ii} as outputs, where *C* is the $k \times (k\text{-}1)$ confusion matrix for the classifier *f*.

Once the inputs and outputs are specified, we need to choose a specific DEA formulation. We chose the classic BCC model [2]. In general, the choice of a particular DEA model determines 1) the implicit returns-to-scale properties, i.e., constant returns-to-scale (CRS) or variable returns-to-scale (VRS)[3]; 2) the geometry of the envelopment surface, with respect to which efficiency measurements will be made and 3) the efficient projection, i.e., the inefficient DMU's path to the efficient frontier. Over the past three decades, the DEA literature has developed a variety of formulations. A nice overview of those different formulations can be found in [8]. As argued in [8] a particular DEA model carries with it an implicit choice among the above three aspects.

BCC model is an appropriate choice here for two reasons. First it is piecewise linear as ROC convex hull is. Second its VRS property loses the implicit assumption that all DMUs are operating at an optimal scale, as assumed by CRS. VRS condition is specified by the third constraint ($1' \lambda = 1$) in formula (1) of section 2. This condition guarantees the linearity of the efficient frontier.

Here we propose one method based on the efficiency score derived from the DEA model. Since the inefficient DMUs are dominated by the efficient ones according to DEA formulation, we select and combine only the efficient models. And we refer to the method as *EMO* (efficient models only).

EMO first selects only the efficient models ($\theta = 1$) identified by the DEA model and then applies majority voting [4, 12] among these efficient models. Suppose in total *M* models are built. Denote the efficiency score of model *j* as θ_j. Let *S* be the set of efficient models where $\theta_j = 1$ and let $|S|$ be the number of efficient models, $|S| \leq M$. Suppose model *j*'s prediction on record *i* is *k*, $k \in (1 \ldots K)$, where *K* is the number of classes. We denote this prediction as Y_{ij}, $Y_{ij} \in (1 \ldots K)$. Define a function C_{ijk} to be 1 if $Y_{ij} = k$ and 0 otherwise. Define probability p_{ik} of record *i* being *k* as $\sum_{j \in S} C_{ijk} / |S|$, that is, the probability that the set of $|S|$ models vote *k*. Finally, the majority prediction of these $|S|$ models can be expressed as follows:

$$\overline{Y_i} = \arg\max_{k \in (1,K)} (p_{ik}) \qquad (3)$$

As we know, the performance of a combined classifier relies upon the strength of individual classifiers. By combining only the efficient models, we would expect a performance improvement of the hybrid classifier. EMO enriches the model combination literature along two dimensions:

1) It systematically determines the number of models to combine based on the given classification task. The number of efficient models is determined endogenously. In contrast, some methods suggest choosing 3 to 5 best models exogenously by users, and that often is ad-hoc [6].
2) It provides a new way of model selection - determining what models to be combined. We introduce the concept "efficient model" as a new approximant to "good" models. Traditionally, classifiers are measured based on variants of prediction error, which is a summarization of a confusion matrix. Efficient models are measured based on the full confusion matrix. By using more information (than prediction error), efficiency scores may characterize model performance better.

[3] CRS and VRS are economic concepts. CRS means the same additional amount of inputs will produce the same proportional additional amount of outputs. While in VRS cases, the proportion might change.

Table 5.1 Comparison of the out of sample mean squared error across datasets

	Number of Records	Number of Classes	Number of Models	Number of Efficient Models	Average Efficiency	Un-weighted Average (UWA)	Variance Based Weighting (VBW)	Efficient Models Only (EMO)
breast-w	333	2	29	2	0.535	0.033	**0.027**	**0.027**
colic	188	2	29	4	0.710	**0.205**	0.223	0.218
credit-a	316	2	29	2	0.615	0.155	**0.149**	0.215
credit-g	487	2	29	2	0.878	0.263	0.279	**0.261**
diabets	360	2	29	4	0.635	0.284	0.275	**0.253**
heart-c	142	5	29	28	0.997	**0.183**	0.211	**0.183**
hear-h	147	5	29	28	0.998	**0.190**	0.197	**0.190**
iris	78	3	29	23	0.991	0.064	**0.051**	0.064
labor	32	2	29	3	0.104	0.281	0.094	**0.063**
lymph	74	4	29	15	0.971	**0.162**	0.189	**0.162**
sick	1877	2	29	2	0.517	0.018	**0.014**	0.036
sonar	103	2	29	3	0.405	**0.155**	0.184	0.184
vote	227	2	29	11	0.580	0.077	0.066	**0.057**
BK	8029	4	19	17	0.981	0.392	0.396	**0.364**
BN	3340	2	19	1	0.661	0.194	**0.175**	0.218
expedia	2318	2	19	1	0.647	0.107	**0.088**	0.122
ebay	30562	2	17	1	0.282	0.021	0.014	**0.010**
rental	1347	3	18	12	0.967	**0.436**	0.528	**0.436**
book	7876	3	19	6	0.874	**0.326**	0.353	0.331
hotel	1779	3	19	11	0.959	0.384	0.411	**0.365**
Average	2981	2.7	25.4	8.8	0.715	0.197	0.196	**0.188**

(Note: Highlighted are the minimum MSE of each dataset across the three methods)

5. EXPERIMENTS AND RESULTS

In this section we empirically test if *EMO* performs well. We first describe the experiment setup: datasets used, the classification models chosen and the benchmarks against which the combing methods are compared. Then we present the experimental results.

We test our methods on 20 datasets for classification problems. Thirteen of these were chosen from the 37 datasets publicly available in the Weka repository[4] [15], along with detailed descriptions of each dataset. All these 13 datasets entail classification tasks and were commonly used in machine learning and data mining fields [4, 5]. The other seven are web usage datasets provided by a market data vendor. Each dataset predicts online customers' purchases based on their demographics and online searching history. Each of the 20 datasets is divided into an in-sample dataset for training and an out-of-sample dataset for testing. All the seven web usage datasets have a common split, 40% in-sample and 60% out-of-sample. The thirteen Weka datasets were already divided into in-sample and out of sample datasets in the Weka repository and we use these datasets. The first three columns of Table 5.1 reports the number of classes and the number of out-of-sample records of each dataset.

We then build classification models on the in-sample datasets. All the models are from Weka's classifier class, which implements most of the popular classifiers including decision trees (J48 algorithm), neural networks, support vector machines, naïve Bayes, Stumps, DecisionTB, Kstar, IBK, OneR, Logistic Regression, IBMDK5, IB1, ZeroR and PARK [15]. In addition, some models have small variants based on different parameterization of the model (e.g. Algorithm J48Part and J48-x5 are both variants of algorithm of J48.) We consider them as different models. We attempted to make use of all these available models. However, for some of the bigger datasets it is computationally expensive to build models such as neural networks. As such, we built 29 models for smaller datasets (size <1000) and fewer models for those bigger ones. The actual number of models we built for each dataset (between 17 and 29) is presented in column 4 of Table 5.1.

After the models are built, we then apply two common model-combining methods as the benchmarks. The first one is the simple un-weighted average method and the second one is a weighted

[4] These datasets are originally maintained by the UCI repository for machine learning [5] and reformatted into Weka format by [15]. Weka is a popular and open-source data mining package that implements most current data mining algorithms in Java.

average method. Un-weighted average (*UWA*) simply averages the predictions made by each model (same as majority vote for classification tasks). It is a commonly used benchmark and has been found to be robust and accurate as compared to other combining methods [1, 6, 12]. The second benchmark (*VBW*) is a weighted average method that combines forecasts inversely proportional to the prediction variance [6].

The DEA software we use is DEAP 2.1 developed by [7]. For each classifier of each dataset, we input the confusion matrix to the DEA Model. In order to conform to the TP/FP convention, here each element in the confusion matrix represents a row percentage. DEAP 2.1 then reports efficient scores for each classifier. Finally we build *EMO*.

We present the results in Table 5.1. Columns 2-6 are summary statistics of each dataset including the number of records (out-of-sample), the number of classes, the number of models applied to that dataset, the number of efficient models identified by DEA and the average efficiency scores across all models. The last three columns represent the three different combining methods: the first two are the two benchmark methods (*UWA* and *VBW*); and the last one is our proposed method - *EMO*. The values of these three columns are the mean squared errors (MSE) in the out-of-sample data.

Based on a simple count, the best combining method is *EMO* (13 out of 20 datasets), followed by *UWA* (7/20) and *VBW* (6/20). Note that for quite a few datasets there are "ties" in the sense that multiple combining approaches are equally good. The average MSE these three methods are 0.188, 0.197 and 0.196 respectively. Further note that *EMO* uses far fewer models (8.8 vs. 25.4 on average) in combination and it still outperforms the two benchmark methods. These results suggest that DEA can be a promising method for model combination. Further this approach is attractive given the availability of tools [7] to easily implement DEA for combing models.

6. CONCLUSIONS

In this paper we presented a DEA approach to combine models. We proved that for the 2-class classification problems, DEA models identify the same convex hull as the popular ROC analysis. We further develop a DEA-based method to combine *k*-class classifiers. Experiments using 29 classifiers on 20 datasets suggest that the method outperforms two other benchmark methods for model combination. Further, we present a detailed literature review and based on this review identify the broader areas in which DEA contributes to the model combination literature.

In future work we plan to further investigate the value of a DEA-based approach for model combination. In particular we plan to study additional weighting schemes and to also use the method to combine multiple models learned from a given classifier.

7. REFERENCES

[1] Ali, K. M., M. J. Pazzani. 1996. Error reduction through learning multiple descriptions. Machine Learning, 24(3): 173-202.

[2] Banker, R. D., A. Chanes, W.W. Cooper. 1984. Some Models for estimating Technical and Scale Inefficiencies In Data Envelopment Analysis. Management Science, 30(9): 1078-1092.

[3] Barber, D., P. Dobkin, H. Huhdanpaa. 1996. The Quickhull Algorithm for Convex Hulls. ACM Transactions on Mathematical Software, 22(4): 469-483.

[4] Bauer, E., Kohavi, R. 1999. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. Machine Learning, 36(1): 105-142.

[5] Blake, C., C, Merz. 1998. UCI Repository of Machine Learning Database: Univ. of California, Department of Information and Computer Science.

[6] Clemen, R. T. 1989. Combining Forecasts: A review and annoted bibliography. International Journal of Forecasting, 5: 559-583.

[7] Coelli, T. 1996. A Guide to DEAP Version 2.1: A Data Envelopment Analysis (Computer) Program: University of New England, Amidale.

[8] Cooper, W., Lawrence Seiford, Kaoru Tone. 2002. Data envelopment analysis: A comprehensive text with models, applications, references and DEA-solver software. Dordrecht, Netherlands: Kluwer Academic Publishers.

[9] Dietterich, T. 2000. Ensemble Methods in Machine Learning. Multiple Classifier Systems. 18, 1-15.

[10] Kim, Y., J. Kim, J. Jongwoo. 2002 Convex hull machine for regression and classification. IEEE Conference on Data Mining 2002. 243-253.

[11] Lane, T. 2000. Position Paper: Extensions of ROC Analysis to Multi-Class Domains. Paper presented at the Proceedings of ICML-2000 Workshop on Cost-Sensitive Learning, Stanford.

[12] Opitz, D. 1999. Popular Ensemble Methods: An Empirical Study. Journal of Artificial Intelligence Research, 11: 169-198.

[13] Provost, F. and T. Fawcett. 2001. Robust Classification for Imprecise Environment. *Machine Learning* 42, 203-231.

[14] Srinivasan, A. 1999. Note on the Location of Optimal Classifiers in n-Dimensional ROC Space: Oxford University.

[15] Witten, I., E. Frank. 1999. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufman.