This article was downloaded by: [144.214.88.242] On: 18 September 2017, At: 20:07
Publisher: Institute for Operations Research and the Management Sciences (INFORMS)
INFORMS is located in Maryland, USA

INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Constructing Ensembles from Data Envelopment Analysis

Zhiqiang Zheng, Balaji Padmanabhan,

To cite this article:
Zhiqiang Zheng, Balaji Padmanabhan, (2007) Constructing Ensembles from Data Envelopment Analysis. INFORMS Journal on Computing 19(4):486-496. https://doi.org/10.1287/ijoc.1060.0180

Full terms and conditions of use: http://pubsonline.informs.org/page/terms-and-conditions

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2007, INFORMS

Please scroll down for article—it is on subsequent pages

INFORMS is the largest professional society in the world for professionals in the fields of operations research, management science, and analytics.
For more information on INFORMS, its publications, membership, or meetings visit http://www.informs.org

RIGHTSLINK

INFORMS Journal on Computing
Vol. 19, No. 4, Fall 2007, pp. 486–496
ISSN 1091-9856 | EISSN 1526-5528 | 07 | 1904 | 0486

DOI 10.1287/ijoc.1060.0180
© 2007 INFORMS

Constructing Ensembles from Data Envelopment Analysis

Zhiqiang Zheng
School of Management, University of Texas at Dallas, Richardson, Texas 75083,
ericz@utdallas.edu

Balaji Padmanabhan
The Wharton School, University of Pennsylvania, Philadelphia, Pennsylvania 19104,
balaji@wharton.upenn.edu

Using an ensemble of models often results in better performance than using a single "best" model. We present a new approach based on data envelopment analysis (DEA) for model combination. We prove that for two-class classification problems, DEA models identify the same convex hull as does the popular receiver operating characteristics (ROC) analysis used for model combination. We further develop two DEA-based methods to combine classifiers for the more general *k*-class classification problems. Our results demonstrate that the two methods outperform other benchmark methods and suggest that DEA can be a powerful tool for model combination.

Key words: model combination; ensembles; data envelopment analysis
History: Accepted by Amit Basu, Area Editor for Knowledge and Data Management; received July 2004; revised May 2005, December 2005; accepted January 2006. Published online in *Articles in Advance* July 20, 2007.

1. Introduction

The idea of combining models to make predictions has been successful for problems in a variety of fields including management science, statistics, data mining, and machine learning (Domingos 1998, Provost and Fawcett 2001, Dietterich 2000, Breiman 1998, Meade and Islam 1998, Russell and Adam 1987, Mojirsheibani 1999, Zhu et al. 2002). Studies have shown, both empirically (Bauer and Kohavi 1999, Opitz and Maclin 1999) and theoretically (Mojirsheibani 1999; Breiman 1996a, 2001), that an ensemble of models consistently outperforms individual models for classification tasks. However, most methods for model combination are still ad hoc.

Provost and Fawcett (2001) developed the well-known ROC (receiver operating characteristics) approach for combining classifiers. A ROC space is specified by two dimensions—the true positive rate (TP) and the false positive rate (FP) of the classifiers. This approach was shown to yield a robust hybrid classifier that outperforms all the individual classifiers. However, as they point out, a limitation of the approach is that ROC analysis can only deal with two-class (binary) problems. While Srinivasan (1999) and Lane (2000) point out that ROC analysis can theoretically be extended to *k*-dimensional classification problems, specific implementations still need to be developed.

We develop a method based on data envelopment analysis (Charnes et al. 1978) to combine models for the more general *k*-class classification problems. DEA addresses the issue of determining the efficiency of various "decision making units" (DMUs) based on how they convert inputs into outputs (Banker et al. 1984, Charnes et al. 1997). In a DEA data set each DMU corresponds to a data point, the attributes of which can be grouped into inputs and outputs. DEA uses a mathematical-programming approach to determine how efficient each DMU is in converting inputs into outputs. The points lying on the convex hull formed by the data are called *efficient* DMUs. DEA provides algorithms to identify efficient DMUs and measures the efficiency of all DMUs against these efficient ones. Since DEA deals with multiple inputs and multiple outputs, this presents a natural opportunity to use the technique to combine models for the general *k*-class case. However, how exactly the model-combination problem translates into a DEA formulation is nontrivial.

We show how the model-combination problem can be formulated as a DEA problem. We prove theoretically that for a two-class problem, the convex hull identified by the ROC analysis is equivalent to the efficient frontier identified by the DEA-based approach. More important, we show how the DEA-based method can extend the ROC-based approach to

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:07 . For personal use only, all rights reserved.

RIGHTSLINK

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:07 . For personal use only, all rights reserved.

k-class problems. Based on extensive experimentation we show that the DEA-based methods for model combination outperform classical benchmark methods.

We first review literature on combining models. After discussing the ROC-based and the DEA-based approaches in Section 2, we show their equivalence for the two-class classification problems in Section 3. Section 4 presents a DEA-based method to deal with the multi-class problem. We report experimental results in Section 5 and conclude in Section 6.

2. Literature Review

Below we review the literature on combing models in Section 2.1 and provide a brief review of DEA to motivate our approach in Section 2.2.

2.1. Combining Multiple Models

The problem of combining models has been addressed across a variety of fields including management science, statistics, forecasting and machine learning. In this section, we review this body of literature around the following two key questions:

- Why should models be combined?
- Given that model combination is useful, how should the multiple models be generated, selected, and combined?

Below we discuss the ideas proposed in the literature in the context of the above questions.

2.1.1. Reasons for Model Combination. As suggested by the "no-free-lunch" theorem (Wolpert and Macready 1997), it is well known that no single model works best for all data sets. Different data characteristics such as data size, normality, linearity, and correlation all affect model performance considerably (Kiang 2003). Even for the same data set, different models may excel in different aspects (Mojirsheibani 1999). Hence, building multiple models and combining them has been found to be effective for a variety of problems.

Building more accurate models is a primary reason for combining models. According to Hansen and Salamon (1990), a necessary and sufficient condition for a combined method to be more accurate than any of its individual members is that the individual methods be accurate and diverse. According to their definition, an accurate model is one that has an error rate better than random guessing; and two models are considered diverse if they make different errors on predictions. The first condition, the accuracy of models, is understandably desirable. The second condition, the diversity of models, is important. The intuition is that if the models are diverse, uncorrelated errors made by the individual models can be removed by combining them (Dietterich 1997). When errors are positively correlated, the incremental benefit from additional predictions (of individual models) may be small (Morrison and Schmittlein 1991, Rust et al. 1995). For decision trees, Breiman (2001) shows that the accuracy of a random forest (combined trees) depends on the strength of the individual tree classifiers and a measure of the dependence between them. However, how to measure "diversity" of classifiers and how to generate diverse classifiers remain open questions. Kuncheva and Whitaker (2003) examined four different measures of diversity and their empirical study did not lend strong support for using diversity in real applications.

Dietterich (1997) provides three reasons for combining multiple models. The first reason is statistical. A learning algorithm can be viewed as searching a space of hypotheses to identify the best hypothesis in the space. If the data size is small, the learning algorithm cannot distinguish between candidate hypotheses, each of which may give the same accuracy on the training data. By constructing an ensemble the algorithm can average their votes and reduce the risk of choosing the wrong classifier. The second reason is computational. Many learning algorithms work by performing some form of local search that may get stuck in local optima (e.g., the greedy splitting rules used in building decision trees). An ensemble constructed by running the local search from many different starting points may provide a better approximation to the true unknown function than do any of the individual classifiers. The third reason is related to model representation. In most applications, the true function cannot be represented by any of the hypotheses. By forming weighted sums of hypotheses, it may be possible to expand the space of the hypothesis functions.

Yet another point of view was put forth by Breiman (1998), who argued that combined models work best for unstable models, where a small perturbation in the training sets may result in large changes in prediction. Modeling techniques such as subset-selection methods in regression, decision trees, and neural networks are unstable. In contrast, linear regression, logistic regression, and *k*-nearest neighbors are considered to be stable.

Given that it may be useful to combine models, how exactly should this be done? The process may be viewed as having three steps: generating multiple models, selecting a subset of models to combine, and combining the selected models. In Sections 2.1.2–2.1.4 we summarize the key ideas as they relate to these three steps (although in practice a single model combination-technique may often combine these steps in an algorithm without separating them explicitly).

2.1.2. Generating Multiple Models. Some research holds that the key to successful model combination is to construct individual models whose errors are at least somewhat uncorrelated (Rust et al. 1995,

RIGHTSLINK

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:07 . For personal use only, all rights reserved.

Morrison and Schmittlein 1991, Dietterich 1997). One approach is to construct truly independent models to ensure low correlation of model errors. However, theories on the independence of different models are not readily available. Thus, there is no consensus on how to generate multiple models in the first place.

Most of the related literature in management science primarily starts with a set of chosen models as component models (Meade and Islam 1998, Russell and Adam 1987, Zhu et al. 2002, Kim et al. 2003, Papatla and Zahedi 2002). For example, Meade and Islam (1998) combined a set of 22 different technology-diffusion models including logistic, Gompertz, and Weibull diffusion models. Zhu et al. (2002) considered four different classifiers including linear discrimination, logit models and two different neural networks.

The literature in machine learning and statistics, however, has primarily focused on constructing multiple models by perturbing the training data, or what Breiman (1998) referred to as "P&C" (perturb and combine) methods. In this approach the same learning algorithm is run several times, each time with a different subset of training examples. Some of the popular methods that fall into this category are bagging (Breiman 1996a), arcing (Breiman 1998), boosting (Freund and Shapire 1996), and many variants of ensembles such as committee of learners, classifier fusion, mixture of experts, and consensus theory (Kuncheva and Whitaker 2003). For a more detailed and complete discussion of these methods, see Dietterich (2000), Kittler and Roli (2001), Valentini and Masulli (2002), and Kuncheva (2004).

Another approach for generating multiple models is based on data splitting. Here the training data are split into disjoint subsets, based on which different models are built (Mojirsheibani 1999). A special case is cross validation (LeBlanc and Tibshirani 1996). The training set is randomly divided into multiple, say ten, disjoint subsets. Then multiple training sets (each containing nine subsets) can be constructed by leaving out a different one of these ten subsets. Ensembles constructed in this way are sometimes called *cross-validated committees* (Dietterich 1997). More recently, Kim et al. (2002) developed a convex-hull ensemble machine (CHEM) to construct multiple models sequentially, where the true error of model $k+1$ is perpendicular to the error between this model and an ensemble of the previous k models.

2.1.3. Evaluating and Selecting Individual Models. Most literature combines all models available, but there are alternatives to this. Russell and Adam (1987) suggested combining only three to five "best" models and found that this worked better than combining all available models. In practice, the number of models depends on the actual context, and is often determined empirically. The model-selection literature addresses this problem by proposing a variety of criteria that measure how well the model approximates the real data-generating process (Meade and Islam 1998). Some of the popular model-selection criteria include AIC (Akaike information criteria), BIC (Bayesian information criteria), and ROC. We review only ROC in detail since it is closely related to the DEA method. See Rust et al. (1995), Meade and Islam (1998), and Ishwaran et al. (2001) for further review on other model-selection criteria.

ROC analysis can be used for model selection for two-class classification problems. One class is normally referred to as *positive* and the other *negative*. A particular classification model (i.e., classifier) maps instances to predicted classes. The *true positive rate* (TP) and *false positive rate* (FP) are defined as

$$\mathrm{TP} = \text{positives correctly classified/total positives},$$

$$\mathrm{FP} = \text{negatives incorrectly classified/total negatives}.$$

An ROC space thus is two-dimensional, normally specified by FP as the x-axis and TP as the y-axis. Conversely, a particular classifier is represented by a pair of TP and FP in the ROC space, and a perfect classifier will have $\mathrm{TP}=1$ and $\mathrm{FP}=0$. Note that simply increasing the number of points that are predicted as "1" (for a binary classifier) will increase the TP of the classifier. At the extreme, a classifier predicting a 1 for any point will have $\mathrm{TP}=1$. However this will also increase the FP of the classifier (in the extreme case the FP will also be 1). This observation suggests a TP and FP tradeoff for classifiers—increasing the TP may also result in a higher FP. The convex hull 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. Clearly models inside the convex hull are dominated by the models on the convex hull. Provost and Fawcett (2001) developed an algorithm, ROC convex hull (ROCCH), to identify the best models on the frontier and then used these best models to construct a hybrid classifier. Their approach is also cost-sensitive—users specify a cost matrix associated with the misclassification (e.g., misclassifying 1 as 0 is four times more costly than misclassifying 0 as 1). The hybrid classifier would then consist of the set of points where the iso-slope (implied by the cost matrix) is tangential to the ROC convex hull. They showed that under any target cost and class distributions, a robust classifier will perform at least as well as the best classifier for those conditions.

2.1.4. Combining Multiple Models. Most literature here has focused on combining the outputs of individual models by assigning weights to the outputs.

RIGHTSLINK

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:07 . For personal use only, all rights reserved.

Unweighted voting, majority voting, weighted voting, and model-based weighting are among the popular approaches used. Unweighted voting, a method that averages the outputs of individual models, is accurate and robust (Clemen 1989). In many applications it even turns out to be the best scheme and it is often employed as the benchmark combination method (Meade and Islam 1998). A special form of unweighted voting is the majority vote as done in bagging (Breiman 1996a) and random forests (Breiman 2001). The majority-vote method picks the class that the majority of classifiers predicted as the final output. Weighted voting methods have many forms. Mojirsheibani (1999) points out that most of the rules of model combination are linear, such as a linear combination of the predicted membership provided by individual classifiers, or a linear combination of the class conditional probabilities from all different classifiers. Papatla and Zahedi (2002) construct a hybrid model from a linear combination of a logit model and a neural network. Zhu et al. (2002) proposed a Bayesian method that computes the posterior probability of being in a certain class given multiple classifiers.

Some other interesting weighting schemes have also been developed. Russell and Adam (1987) and Kim et al. (2002, 2003) assign weights that are inversely proportional to the individual errors as measured by the MSE (mean squared error). In AdaBoost, the weight for a classifier is computed from the accuracy of the classifier measured on the weighted training distribution that was used to learn this classifier (Freund and Shapire 1996). Except for some simple situations, it is not always clear how one can most effectively assign weights. Some studies have attempted to use a model to learn the weights. Kim et al. (2002) proposed using a genetic algorithm to learn the weights, where the algorithm searches for a good weight scheme through a fitness function. Ali and Pazzani (1996) described a likelihood-combination method in which they apply the Naïve-Bayes algorithm to learn the weights. Stacking (Breiman 1996b) is a method to form a single final classifier that encompasses the multiple classifiers. The data are first partitioned into disjoint subsets $S_1, S_2, \ldots, S_m$. Each classifier is applied to S_{-i} (all the sets excluding S_i), and predict the labels to get Y_i. Using K classifiers, there are K predictions for subset i. Then a new algorithm (e.g., linear regression, naïve Bayes) is used to learn the relationship between the true label and the predicted labels.

2.2. Data Envelopment Analysis (DEA)

DEA is a nonparametric approach that constructs an efficient frontier (envelopment) over the data, and calculates each data point's efficiency relative to this frontier (Charnes et al. 1997). DEA assumes that the variables in the data can be logically divided into inputs and outputs. Each data point corresponds to a DMU in practice (i.e., for traditional DEA applications). The objective of DMUs is to convert inputs into outputs as efficiently as possible. A DMU is rated as fully efficient (100%) if and only if the performance of other DMUs does not show that some of its inputs or outputs can be improved without worsening some of its other inputs or outputs (Cooper et al. 2004). DEA uses mathematical programming to identify the efficient DMUs. The efficient frontier consists of these efficient units and the efficiencies of all other DMUs are computed with respect to this frontier.

An objective of DEA is to project inefficient DMUs to the efficient frontier in order to understand what needs to be done for the inefficient DMU to move toward efficiency. In input-oriented DEA models the idea is that inefficient DMUs can decrease inputs, keeping the output constant in order to become efficient. Similarly, output-oriented DEA models are used when the inputs are fixed but the outputs of a DMU need to be maximized with respect to the efficient frontier.

The DEA model in its original form—known as the CCR model—represented the performance or efficiency of the DMU as the ratio of weighted outputs to weighted inputs (Charnes at al. 1978). The CCR model assumes constant returns to scale (CRS)—the value of the output variables increase proportionally to the value of the inputs. More generally, return to scale refers to constant, increasing, or decreasing efficiency based on the production scale (Fare et al. 1994). The classic BCC model (Banker et al. 1984) deals with the variable-returns-to-scale (VRS) case. Below, we briefly discuss the formulation of the BCC model.

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 consisting of N DMUs. The columns represent the data points and the rows are the variables. Hence for the ith DMU the inputs are represented by the $K \times 1$ vector x_i and the outputs by the $M \times 1$ vector y_i. The input-oriented and the output-oriented BCC models are:

$$\text{Min}\{\theta \mid -y_i + Y\lambda \geq 0;\ \theta x_i - X\lambda \geq 0;\ 1'\lambda = 1;\ \lambda > 0\}, \quad (1)$$

$$\text{Max}\{\theta \mid -\theta y_i + Y\lambda \geq 0;\ x_i - X\lambda \geq 0;\ 1'\lambda = 1;\ \lambda > 0\}. \quad (2)$$

In (1) and (2), 1 represents an $N \times 1$ vector of ones, θ is a scalar, and λ is an $N \times 1$ vector of constants. In the model, θ and λ are the only two variables to be optimized. All the other values (X, Y) are given by the data. This linear-programming formulation is used essentially to identify a convex hull for all the data points and to compute an efficiency score θ for

RIGHTSLINK

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:07 . For personal use only, all rights reserved.

each individual data point. The value of θ is bounded between 0 and 1, with 1 indicating a point on the frontier. Note that this must be solved N times, once for each DMU in the sample.

There have been many applications of DEA and detailed references can be found in Fare et al. (1994), Charnes et al. (1997), Cooper et al. (2004), and Coelli (1996). Some applications of DEA include studying the efficiency of commercial banks (Seiford and Zhu 1999, Soteriou and Zenios 1999), anticipating the consequences of school reforms (Grosskopf et al. 1999) and understanding online customers' shopping efficiency (Xue and Harker 2002). This paper, however, represents the first attempt to apply DEA to study model efficiency and model combination.

3. Equivalence of the DEA and ROC Convex Hulls

Both ROCCH and DEA (described in Sections 2.1 and 2.2, respectively) identify a convex hull. In this section we show that the two convex hulls are identical for a two-class classification problem.

Consider the standard learning problem. A learning algorithm is given training examples of the form (x, y) for some unknown function $y = f(x)$. We restrict attention to discrete y values, i.e., we focus on classification problems and the models considered here are classifiers. The y values are from a discrete set of classes $C \in (1, \ldots, K)$. Given a set of training examples, a learning algorithm outputs a classifier. The classifier is a hypothesis about the true function f. When there exists a set of models, $f_1, f_2, \ldots, f_m$, a model-combination method h is a model of the form $y = h(f_1(x), f_2(x), \ldots, f_m(x))$.

3.1. ROC Convex Hull Algorithm

ROC convex hull (Provost and Fawcett 2001) has several distinct characteristics:

1. It is two-dimensional, characterized by the TP and the FP of a classifier.
2. It goes through $(0, 0)$ and $(1, 1)$—when a naïve model classifies all instances as negative or positive, respectively—and both TP and FP 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 (Barber et al. 1996). The computation complexity for this test is $O(N^4)$ for N records. ROCCH uses an algorithm similar to the quickhull algorithm (Provost and Fawcett 2001, Barber et al. 1996), the complexity of which is reduced to $O(N \log N)$. The property of monotonicity further simplifies the core of the algorithm summarized literally (Provost and Fawcett 2001, p. 214) below.

> 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.

3.2. DEA Convex Hull

It is not obvious how to represent an ROC problem in a DEA framework. Here we take the approach that a classifier can be considered as 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 in a DEA framework.

Definition 1. A two-class classifier $\{f(x, y), y \in (0, 1)\}$ is a DMU in DEA that takes FP as inputs and TP as outputs.

Once the inputs and outputs are specified, we need to choose a specific DEA formulation. Over the past two decades, the DEA literature has developed a variety of formulations. 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), (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. Charnes et al. (1997) maintain that a particular DEA model carries with it an implicit choice among the above three aspects.

We choose the classic BCC model (Banker et al. 1984). The BCC model is an appropriate choice here for three reasons. First, it is piecewise linear, as is ROC convex hull. Second, its VRS property relaxes the implicit assumption that all DMUs are operating at an optimal scale, as assumed by CRS, and thus is more general. For classifiers, there is no reason to believe that a proportional reduction in FP will yield the same proportional increase in TP. Third, we choose an input-oriented formulation since we view a classifier as having a specific TP rate; making this more efficient may be done by reducing the FPs of the classifier.

As mentioned in Section 2.2, the BCC formulation below has to be solved for each classifier separately to obtain its efficiency score. Among a set of M classifiers for evaluating a particular classifier i (with

RIGHTSLINK

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:07 . For personal use only, all rights reserved.

FP $= x_i$ and TP $= y_i$), the one-input one-output BCC model simplifies to

$$\text{Min}\left\{\theta \,\middle|\, \sum_{j=1}^{M}\lambda_j y_j \geq y_i;\ \sum_{j=1}^{M}\lambda_j x_j \leq \theta x_i;\ \sum_{j=1}^{M}\lambda_j = 1;\ \lambda_j > 0\right\}. \quad (3)$$

According to Charnes et al. (1997), the ith DMU is on the efficiency frontier if and only if $\theta = 1$. Thus, in order to test whether ROCCH identifies the same convex hull, we need to prove one of the following two cases:

1. A model is on the convex hull of ROCCH if and only if it has the property $\theta = 1$ in the DEA model.
2. A model is *not* on the convex hull of 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 hull as ROCCH for a two-class classification problem. We do this by using the second test in Section 3.2.

Proposition 1. *An inefficient point in the ROC space has efficiency score $\theta < 1$ in DEA.*

Proof. This can be shown graphically (Figure 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, $X_j < x_i < X_{j+1}$, which is below the line in Figure 1 (FP is the x-axis and TP is the y-axis.) Now the test simplifies to whether the value of θ of point (x_i, y_i) is less than one in a DEA model.

It is easy to verify that this is indeed the case. Clearly, (x_i, y_i) satisfies the first condition ($\sum \lambda_j Y_j \geq y_i$) of the BCC model (see formulation (3)), for (x_i, y_i) is below the line, which delineates a linear combination of the two points. The 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 1). According to the property of linear combination, $x_i' = \sum \lambda_j X_j$. Thus $\sum \lambda_j X_j < x_i$ or equivalently $\theta < 1$. □

Proposition 2. *A DMU with $\theta < 1$ is an inefficient point in the ROC space.*

Proof. If a classifier (x_i, y_i) has $\theta < 1$ in the DEA formulation, then $\sum \lambda_j X_j < x_i$ and $\sum \lambda_j Y_j \geq y_i$. This implies that the point (x_i, y_i) is below the two adjacent classifiers (X_j, Y_j) and (X_{j+1}, Y_{j+1}) on the convex hull, where $X_j < x_i < X_{j+1}$ (see Figure 1). □

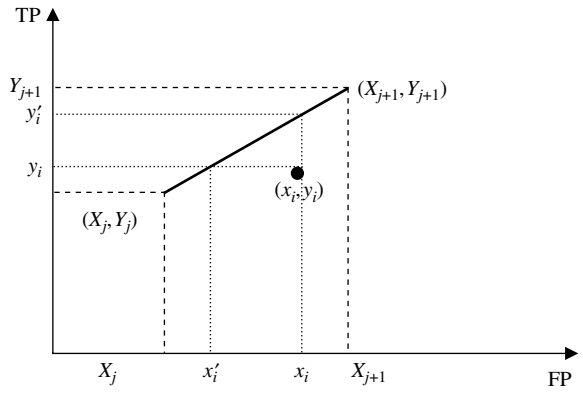

Figure 1 An Inefficient Point in the ROC Space

Based on the equivalence of the two tests described in 3.2, we have:

Corollary 1. *A point on the convex hull of the ROC space has efficiency score $\theta = 1$ in DEA.*

Corollary 2. *A DMU with $\theta = 1$ is an efficient point in the ROC space.*

Thus for two-class problems, the convex hull identified by ROCCH is identical to the one identified by a DEA model.

4. A DEA Approach for Combining k-Class Classifiers

The ability of DEA to deal with multiple inputs and multiple outputs suggests that DEA could extend ROCCH to k-class problems. In this section we develop a DEA approach to combine classifiers for more general multi-class classification tasks. Further, we will show that the DEA-based approach contributes to both the selection and combination steps in model combination.

The first step of a typical DEA application is to identify the inputs and outputs. As discussed for the two-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 inputs and outputs are for the general k-class classification problem. Lane (2000) pointed out that for a k-class problem, a particular classifier could be represented by a $k \times k$ confusion matrix (i.e., misclassification cost matrix). Let C denote the $k \times k$ confusion matrix. Each element c_{ij} represents the number of times that class i is misclassified as j, $i \neq j$ and $i, j \in (1, k)$; and when $i = j$, c_{ii} is the number of times 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}. In DEA terms, a classifier can be viewed as a decision maker

RIGHTSLINK

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:07 . For personal use only, all rights reserved.

that attempts to produce diagonal elements c_{ii} at the expense of off-diagonal elements c_{ij}. In this case, c_{ij} can be interpreted as the inputs and c_{ii} as the outputs in a DEA model. Notice that the actual degrees of freedom of C is k^2-k. This is evident if we remove the k diagonal elements. In our experiments we remove the kth column since we use the diagonal elements as outputs in our DEA formulation.

Definition 2. 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 $k \times (k-1)$ confusion matrix for the classifier f.

Once the inputs and outputs are specified, we need to choose a specific DEA formulation. For reasons mentioned in Section 3.2 we choose the classic BCC formulation here. Although various other DEA formulations may also be useful for model combination, we do not compare alternative DEA formulations in this paper. This needs to be studied in future work.

It is clear that the DEA approach contributes to model selection since DEA partitions classifiers into efficient and inefficient ones. Further, it computes the efficiency scores for each classifier in comparison with the frontier. The efficient ones have $\theta = 1$ and the inefficient ones have $\theta < 1$. Thus the efficiency score θ can be used as a new criterion to select models.

Less clear is how to combine multiple classifiers using DEA. Here we propose two methods, both of which are based on the efficiency scores derived from the DEA model. The first one only combines the efficient models identified by DEA and we refer to it as EMO (*efficient models only*). The second one weighs each model proportional to θ, which we refer to as ESW (*efficiency score weighting*).

4.1. Method EMO—Efficient Models Only

EMO first selects only the efficient models ($\theta = 1$) identified by the DEA procedure and then applies majority voting (Breiman 1996a, Dietterich 1997) 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, so $|S| \le M$. 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 $C_{ijk} = 1$ 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

$$\overline{Y}_i = \arg\max_{k \in \{1,\ldots,K\}}(p_{ik}), \quad \overline{Y}_i \in \{1, \ldots, K\}. \tag{4}$$

As reviewed in Section 2, the performance of a combined classifier relies upon the strength of individual classifiers. By combining only the efficient models, we would expect performance improvement in 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. Hence, the number of efficient models is determined endogenously. This is in contrast to approaches that choose a set of models exogenously by users, often in an ad hoc manner.

2. It provides a new method for model selection. We introduce the concept of "efficient models" as a new approximation for "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. This leads to our second method, ESW.

4.2. Method ESW—Efficiency Score Weighting

Unlike EMO that only combines efficient models, ESW uses all M models. In particular, ESW weighs each model according to its efficiency score θ_j as follows:

$$\overline{Y}_i = \arg\max_{k \in \{1,\ldots,K\}}\left(\sum_j \theta_j C_{ijk}\right), \quad \overline{Y}_i \in \{1, \ldots, K\}. \tag{5}$$

The use of a linear weighting scheme here is related to our modeling objective. We assume our objective is to minimize the expected generalization error, say, the expected mean squared error (MSE) in out-of-sample data. For the k-class classification problems, a classifier's prediction accuracy is $r = \sum_{i \in \{1,\ldots,K\}} C_{ii}/N$, where C_{ii} is a diagonal element in the confusion matrix C and N is the total number of records. Notice that here we assume each element in the confusion matrix represents the number of misclassifications. Then for classification problems, we have $\text{MSE} = 1 - r$ (i.e., this is the same as a 0-1 loss function since any error in such problems is either a zero or one). So the MSE has a positive linear relationship with C_{ij}, $i \ne j$. Other things equal, increasing C_{ij} leads to an increase in MSE. The efficiency score θ_j is derived from a DEA model built on the confusion matrix on the in-sample data. If we assume the same distribution in the training and testing data, then we should expect a negative (i.e., more efficient models lead to fewer misclassifications) linear relationship between θ and misclassification errors C_{ij}. Hence a linear scheme is used.

5. Experiments

In this section we empirically test if the two proposed DEA-based methods, EMO and ESW, perform well. We first describe the data sets used, the classification models chosen, and the benchmarks against

RIGHTSLINK

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:07 . For personal use only, all rights reserved.

Table 1 Summary Results of the Experiments

Data set	Number of records	Number of classes	Number of models	Number of efficient models	Average efficiency	Unweighted average (UWA)	Variance based weighting (VBW)	Efficient models only (EMO)	Efficiency score weighting (ESW)
Breast-w	333	2	29	2	0.535	0.033	**0.027**	**0.027**	0.030
Colic	188	2	29	4	0.710	0.205	0.223	0.218	**0.165**
Credit-a	316	2	29	2	0.615	0.155	0.149	0.215	**0.139**
Credit-g	487	2	29	2	0.878	0.263	0.279	0.261	**0.255**
Diabetes	360	2	29	4	0.635	0.284	0.275	0.253	**0.242**
Heart-c	142	5	29	28	0.999	**0.450**	0.550	**0.450**	**0.450**
Heart-h	147	5	29	28	0.998	**0.329**	0.348	**0.329**	**0.329**
Iris	78	3	29	23	0.991	0.064	**0.051**	0.064	0.064
Labor	32	2	29	3	0.104	0.281	0.094	**0.063**	**0.063**
Lymph	74	4	29	15	0.971	**0.162**	0.189	**0.162**	**0.162**
Sick	1,877	2	29	2	0.517	0.018	**0.014**	0.036	0.017
Sonar	103	2	29	3	0.405	0.155	0.184	0.184	**0.146**
Vote	227	2	29	11	0.580	0.077	0.066	**0.057**	**0.057**
BK	8,029	4	19	17	0.981	0.392	0.396	0.364	**0.361**
BN	3,340	2	19	1	0.661	0.194	**0.175**	0.218	0.186
Expedia	2,318	2	19	1	0.647	0.107	**0.088**	0.122	0.103
Ebay	30,562	2	17	1	0.282	0.021	0.014	**0.010**	0.011
Rental	1,347	3	18	12	0.967	0.434	0.528	0.436	**0.428**
Book	7,876	3	19	6	0.874	**0.326**	0.353	0.331	**0.326**
Hotel	1,779	3	19	11	0.959	0.384	0.411	**0.365**	0.381
Average	2,981	2.7	25.4	8.8	0.715	0.217	0.220	0.208	0.196

Notes. Boldface denotes the minimum MSE of each data set across the four methods. Note that there are equal values among each data set's minimum values.

which the combining methods are compared. Then we present the experimental results.

We test our methods on 20 data sets, thirteen of which were chosen from the 37 data sets publicly available in the Weka repository (Witten and Frank 1999), which is a popular open-source data-mining platform that implements most current data-mining algorithms in Java. All 13 of these data sets were commonly used for classification tasks (Bauer and Kohavi 1999, Blake and Merz 1998). The other seven are Web-usage data sets provided by a market data vendor, in which each data set contains information on online customer purchases, demographics, and Web-browsing history. All 13 data sets in the Weka repository are pre-divided into an in-sample data set for training and an out-of-sample data set for testing. All 7 Web usage data sets were partitioned such that 40% of the data were used as training data and the remaining 60% were used as out-of-sample data. Table 1 reports the number of classes and the number of out-of-sample records in each data set. A limitation of the results is that they are all based on a single holdout sample for each data set. However, this permits easier comparison to our method since Weka has fixed training and test sets that can be directly used by any approach.

We then build classification models on the training data sets. 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 (Witten and Frank 1999). In addition, some models have variants based on different parameterizations of the model (e.g., Algorithm J48Part and J48-x5 are both variants of algorithm of J48) and we consider them as different models here. We attempted to make use of all the available models, but for computational reasons we built 29 models for smaller data sets (size $< 1{,}000$) and fewer models for the bigger data sets. The actual number of models we built for each data set is in column 4 of Table 1.

After the models are built, we apply two common model-combining methods as benchmarks. The first is the simple unweighted-average method, and the second is a popular weighted-average method. Unweighted-average (UWA) simply averages the predictions made by each model. It is commonly used and is robust and accurate compared to other combining methods (Meade and Islam 1998, Winkler and Makridakis 1983). Using the notation in Section 4, the maximum of the average of the predictions of these M models is

$$\overline{Y}_i = \arg\max_{k\in\{1,\ldots,K\}} \left(\sum_j C_{ijk}/M \right). \tag{6}$$

The second benchmark is a well-known weighted-average method that combines forecasts inversely

RIGHTSLINK

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:07 . For personal use only, all rights reserved.

proportional to the prediction (Winkler and Makridakis 1983, Morrison and Schmittlein 1991, Russell and Adam 1987, Kim et al. 2002). Winkler and Makridakis (1983) show that this weighting scheme is optimal when the model objective is to minimize prediction variance. Denote the in-sample variance of model j as σ_j^2, computed as the mean square error of the prediction on the in-sample data. Then the weight is assigned to be c/σ_j^2, where c is a constant such that the total weights is normalized to be 1, so

$$\bar{Y}_i = \arg\max_{k\in\{1,\ldots,K\}}\left(\sum_j c \times C_{ijk}/\sigma_j^2\right). \quad (7)$$

The DEA software we use is DEAP 2.1 developed by Coelli (1996). For each classifier on each data set we input the confusion matrix to the DEA Model. Conforming 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 and ESW.

The results are in Table 1. Columns 2–6 are summary statistics of each data set including the number of records (out of sample), the number of classes, the number of models applied to that data set, the number of efficient models identified by DEA, and the average efficiency scores across all models. The last four columns represent the four different combining methods: the first two are the two benchmark methods (UWA and VBW); and the last two are the two DEA-based methods (EMO and ESW). The values in these four columns are MSE in the out-of-sample data.

Based on a simple count, the best combining method is ESW (13 out of 20 data sets), followed by EMO (8/20), VBW (5/20), and UWA (4/20). The average MSE of these four methods are 0.196, 0.208, 0.220, and 0.217 respectively. Further, paired t-tests across 20 data sets in Table 2 show that ESW significantly outperforms the other three methods (all the p-values are less than 0.05 and t-values are negative which means that the out-of-sample MSE is smaller). EMO is not significantly better than the other two benchmark methods. In addition, there is no significant difference between the two benchmark methods (UWA and VBW).

Table 2 **Paired t-Tests Across the Four Methods**

Method	UWA	VBW	EMO	ESW
UWA	—			
VBW	0.37 (0.32)	—		
EMO	0.24 (−0.71)	0.08 (−1.40)	—	
ESW	**0.03** (−1.94)	**0.001** (−3.47)	**0.01** (−2.54)	—

Notes. The values are p-values with t-values in parentheses, reading from row to column. Boldface denotes the significant p-values based on a one-way test.

As the results demonstrate, the two DEA-based methods perform better than the common benchmarks. Note that EMO uses far fewer models (8.8 vs. 25.4 on average) in combination and yet its results are still comparable to the two benchmark methods. These results suggest that DEA is an effective method for model combination.

However, there is a practical constraint in using DEA for model combination when the number of classes is large. As Andersen and Petersen (1993) point out, a weakness of DEA is that a considerable number of observations typically is characterized as efficient, *unless* the sum of the number of inputs and outputs is small relative to the number of observations. In our experiment, we consider up to 29 different models for most data sets. While these 29 models represent a relatively complete set of classifiers available, they are not adequate to observe the effect of how the DEA-based method performs for a large number of classes, due to the size of the inputs and outputs. As clear from Definition 2, a k-class problem needs $k-1$ outputs and $(k-1)^2$ inputs in DEA. Ideally, the number of models built should be large compared to $O(k^2)$.

In our case, therefore, beyond $k=5$ it is unlikely to get performance improvement using the DEA approach. While there is little theory on how the number of inputs and outputs (corresponding to k) affects DEA's performance, in practice DEA labels more models as efficient as the size of the inputs and outputs grow. This can also be seen empirically. We varied k for the same data set (by grouping categories together) to see how it affects the performance of ESW on the same data set. In our experiments we have two data sets that have five classes (Heart-c and Heart-h). We combined the minority classes and created new data sets with four, three and two classes respectively. This enables us to observe how k affects the relative performance of ESW over UWA on the same data set. The result is reported in Table 3.

Table 3 shows that as k increases, the performances of both ESW and UWA decrease. This indicates that the multi-class problem is harder. Also, the relative gain of ESW over UWA decreases, primarily due to the fact that DEA identifies almost all the models as efficient due to the size of the inputs and outputs being comparable to the number of observations.

Clearly, one solution is to build many models, perhaps by using the P&C methods described in Section 2. While the DEA-based approach presents a general framework that can be used to combine classifiers for the general k-class problem, in practice we need to combine our approach with model-generation methods that generate a large number of (different) models for the DEA-based method to be effective for a truly large number of classes. While we do not

RIGHTSLINK

Table 3 Relative Performance of ESW over UWA as k Increases on the Same Data Set

Data set	Number of classes	Number of efficient models	Average efficiency	Unweighted average (UWA)	Variance based weighting (VBW)	Efficient models only (EMO)	Efficiency score weighting (ESW)	Gain of ESW over UWA (%)
Heart-c	2	6	0.652	0.183	0.183	0.190	**0.148**	19.2
	3	17	0.973	0.471	0.386	0.373	**0.359**	23.6
	4	23	0.984	0.469	**0.456**	**0.456**	**0.456**	2.9
	5	28	0.999	**0.450**	0.550	**0.450**	**0.450**	0.0
Hear-h	2	2	0.63	0.197	0.190	0.190	**0.177**	10.3
	3	20	0.967	0.315	**0.288**	0.295	**0.288**	8.7
	4	27	0.994	0.347	0.345	**0.338**	**0.338**	2.6
	5	28	0.998	**0.329**	0.348	**0.329**	**0.329**	0.0
Average				0.345	0.343	0.328	0.318	

Note. Boldface denotes the minimum MSE of each data set across the four methods.

address this in this paper, this is an excellent area for future work.

6. Conclusion

This paper presents a DEA-based approach for combining models. We prove that for the two-class classification problems, DEA procedures identify the same convex hull as does the popular ROC analysis. We further develop two DEA-based methods to combine k-class classifiers for the general case. Experiments demonstrate that the two methods outperform two other benchmark methods and suggest that DEA-based approaches can be promising for the important and active research area of model combination. More generally, in the spirit of prior work (Mangasarian 1997, Olafsson and Yang 2002, Padmanabhan and Tuzhilin 2003) that demonstrates how OR-based methods can be used in data mining, our work here suggests yet another specific problem for which OR methods can contribute to the area of data mining.

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 in a DEA framework, by identifying the inputs and outputs for DEA based on the misclassification matrix of classifiers. Our method can be used to combine the more general k-class classification problems, extending the ROCCH approach (Provost and Fawcett 2001). We demonstrate empirically that model combination using DEA outperforms conventional approaches for a variety of classification problems.

As mentioned above, there are important problems for future research that can further provide guidance on how DEA-based approaches can best be used. One such problem is exhaustively investigating alternative DEA formulations and understanding the relative advantages. A second problem is studying how model-generation methods can be optimally integrated with the DEA-based approach for large data sets and a large number of classes.

Acknowledgments

The authors thank Area Editor Amit Basu and three anonymous reviewers for several suggestions that substantially improved this paper. They also thank participants at the *Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD2004*) for their helpful comments. The current affiliation of Balaji Padmanabhan is the Department of Information Systems and Decision Sciences, College of Business Administration, University of South Florida, Tampa, Florida 33620; e-mail: bpadmana@coba.usf.edu.

References

Ali, K. M., M. J. Pazzani. 1996. Error reduction through learning multiple descriptions. *Machine Learn.* **24** 173–202.

Andersen, P., N. Petersen. 1993. A procedure for ranking efficient units in data envelopment analysis. *Management Sci.* **39** 1261–264.

Banker, R. D., A. Chanes, W. W. Cooper. 1984. Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management Sci.* **30** 1078–1092.

Barber, D., P. Dobkin, H. Huhdanpaa. 1996. The quickhull algorithm for convex hulls. *ACM Trans. Math. Software* **22** 469–483.

Bauer, E., R. Kohavi. 1999. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learn.* **36** 105–142.

Blake, C., C. Merz. 1998. UCI repository of machine learning database. Technical report, Department of Information and Computer Science, University of California, Irvine, CA.

Breiman, L. 1996a. Bagging predictors. *Machine Learn.* **24** 123–140.

Breiman, L. 1996b. Stacked regressions. *Machine Learn.* **24** 49–64.

Breiman, L. 1998. Arcing classifiers. *Ann. Statist.* **26** 801–849.

Breiman, L. 2001. Random forests. *Machine Learn.* **45** 5–32.

Charnes, A., W. W. Cooper, E. Rhodes. 1978. Measuring the efficiency of decision making units. *Eur. J. Oper. Res.* **2** 429–444.

Charnes, A., W. Cooper, A. Lewin, L. Seiford. 1997. *Data Envelopment Analysis, Theory, Methodology and Applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Clemen, R. T. 1989. Combining forecasts: A review and annoted bibliography. *Internat. J. Forecasting* **5** 559–583.

Coelli, T. 1996. A guide to DEAP version 2.1: A data envelopment analysis (computer) program. Technical report, Department of Operations Research, University of New England, Armidale, New South Wales, Australia.

Cooper, W., L. Seiford, J. Zhu. 2004. *Handbook on Data Envelopment Analysis*. Kluwer Academic Publishers, Boston, MA.

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:07 . For personal use only, all rights reserved.

RIGHTSLINK

Dietterich, T. 1997. Machine learning research: Four current directions. *AI Magazine* **18** 97–136.

Dietterich, T. 2000. Ensemble methods in machine learning. *Multiple Classifier Systems* **18** 1–15.

Domingos, P. 1998. Knowledge discovery via multiple models. *Intelligent Data Anal.* **2** 187–202.

Fare, R., S. Grosskopf, C. A. K. Lovell. 1994. *Production Frontiers*. Cambridge University Press, West Nyack, NY.

Freund, Y., R. E. Shapire. 1996. Experiments with a new boosting algorithm. *Proc. Thirteenth National Conf. Machine Learn.* 148–156.

Grosskopf, S., J. Kathy, L. Hayes, W. Taylor, L. Weber. 1999. Anticipating the consequences of school reform: A new use of DEA. *Management Sci.* **45** 608–620.

Hansen, L., P. Salamon. 1990. Neural network ensembles. *IEEE Trans. Pattern Anal. Machine Intelligence* **12** 993–1001.

Ishwaran, H., L. F. James, J. Sun. 2001. Bayesian model selection in finite mixtures by marginal density decompositions. *J. Amer. Statist. Assoc.* **96** 1316–1332.

Kiang, Y. M. 2003. A comparative assessment of classification methods. *Decision Support Systems* **35** 441–454.

Kim, E., W. Kim, Y. Lee. 2003. Combination of multiple classifiers for the customer's purchase behavior prediction. *Decision Support Systems* **34** 167–175.

Kim, Y., J. Kim, J. Jongwoo. 2002. Convex hull machine for regression and classification. *IEEE Conf. Data Mining, Maebashi City, Japan*, 243–253.

Kittler, J., F. Roli. 2001. Multiple classifier systems. *Proc. 2nd Internat. Workshop on MCS*, Cambridge, UK. Springer-Verlag, Berlin, Germany.

Kuncheva, L. 2004. *Combining Pattern Classifiers*: *Methods and Algorithms.* Wiley, New York.

Kuncheva, L., C. Whitaker. 2003. Measures of diversity in classifier ensembles. *Machine Learn.* **51** 181–207.

Lane, T. 2000. Position paper: Extensions of ROC analysis to multiclass domains. *Proc. ICML-2000 Workshop on Cost-Sensitive Learn.*, Stanford University, Palo Alto, CA.

LeBlanc, M., R. Tibshirani. 1996. Combining estimates in regression and classification. *J. Amer. Statist. Assoc.* **9** 1641–1650.

Mangasarian, O. L. 1997. Mathematical programming in data mining. *Data Mining and Knowledge Discovery J.* **1** 183–210.

Meade, N., T. Islam. 1998. Technological forecasting: Model selection, model stability, and combining models. *Management Sci.* **44** 1115–1130.

Mojirsheibani, M. 1999. Combining classifiers via discretization. *J. Amer. Statist. Assoc.* **94** 600–609.

Morrison, D. G., D. C. Schmittlein. 1991. How many forecasters do you really have?: Mahalanobis provides the intuition for the surprising Clemens and Winkler result. *Oper. Res.* **39** 519–523.

Olafsson, S., J. Yang. 2002. Scalable optimization-based feature selection. *Proc. SIAM Workshop Discrete Math. Data Mining, Arlington, VA*, 53–64.

Opitz, D., R. Maclin. 1999. Popular ensemble methods: An empirical study. *J. Artificial Intelligence Res.* **11** 169–198.

Padmanabhan, B., A. Tuzhilin. 2003. On the use of optimization for data mining: Theoretical interactions and eCRM opportunities. *Management Sci.* **49** 1327–1342.

Papatla, P., M. Zahedi. 2002. Leveraging the strengths of choice models and neural networks: A multi-product comparative analysis. *Decision Sci.* **33** 433–468.

Provost, F., T. Fawcett. 2001. Robust classification for imprecise environment. *Machine Learn.* **42** 203–231.

Russell, D. T., E. E. Adam. 1987. An empirical evaluation of alternative forecasting combinations. *Management Sci.* **33** 1267–1276.

Rust, T. R., D. Simester, R. J. Brodie, V. Nilikant. 1995. Model selection criteria: An investigation of relative accuracy, posterior probabilities, and combination of criteria. *Management Sci.* **41** 322–333.

Seiford, M. L., J. Zhu. 1999. Profitability and marketability of the top 55 U.S. commercial banks. *Management Sci.* **45** 1270–1288.

Soteriou, A., S. A. Zenios. 1999. Operations, quality, and profitability in the provision of banking services. *Management Sci.* **45** 1221–1238.

Srinivasan, A. 1999. Note on the location of optimal classifiers in n-dimensional ROC space. Technical report, Department of Computer Science, Oxford University, Oxford, UK.

Valentini, G., F. Masulli. 2002. Ensembles of learning machines. *Lecture Notes in Computer Sciences*, Vol. 2486. Springer-Verlag, Berlin, 3–19.

Winkler, R. L., S. Makridakis. 1983. The combination of forecasts. *J. Roy. Statist. Soc.* **146** 150–157.

Witten, I., E. Frank. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufman, San Francisco, CA.

Wolpert, D., W. G. Macready. 1997. No free lunch theorems for optimization. *IEEE Trans. Evolutionary Comput.* **1** 67–82.

Xue, M., P. T. Harker. 2002. Customer efficiency: Concept and its impact on e-business management. *J. Service Res.* **4** 253–267.

Zhu, H., P. Beling, G. Overstreet. 2002. A Bayesian framework for the combination of classifier outputs. *J. Oper. Res. Soc.* **53** 719–727.

Downloaded from informs.org by [144.214.88.242] on 18 September 2017, at 20:07 . For personal use only, all rights reserved.

RIGHTSLINK