Computers & Industrial Engineering 119 (2018) 181–192

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Data envelopment analysis models for probabilistic classification

Parag C. Pendharkar

Information Systems, School of Business Administration, Pennsylvania State University at Harrisburg, 777 West Harrisburg Pike, Middletown, PA 17057, United States

ARTICLE INFO

Keywords:
Data envelopment analysis
Classification problem
Probabilistic classification
Misclassification costs
Neural networks

ABSTRACT

We propose and test three different probabilistic classification techniques using data envelopment analysis (DEA). The first two techniques assume parametric exponential and half-normal inefficiency probability distributions. The third technique uses a hybrid DEA and probabilistic neural network approach. We test the proposed methods using simulated and real-world datasets. We compare them with cost-sensitive support vector machines and traditional probabilistic classifiers that minimize Bayesian misclassification cost risk. The results of our experiments indicate that the hybrid approach performs as well as or better than other techniques when misclassification costs are asymmetric. The performance of exponential inefficiency distribution DEA classifiers is similar or better than that of traditional probabilistic neural networks. We illustrate that there are certain classification problems where probabilistic DEA based classifiers may provide superior performance compared to competing classification techniques.

1. Introduction

Good solutions to classification problems can have a significant impact on organization revenue and profitability. Typical high-impact classification problems in organizational decision-making, are associated with customer risk management (Bose & Chen, 2009) and organization default risk management, among others. A sub-set of the classification problem is the asymmetric misclassification cost classification problem, where the risk of misclassification is not symmetrical. Bult and Wittink (1996) identified three different kinds of misclassification risk, and these three different risks were: symmetric risk where all misclassification costs were equal, asymmetric homogeneous risk where misclassification costs were not equal but were constant, and asymmetric non-homogeneous risk where misclassification costs were not equal but were variable for each case. Zhao, Zhao, and Song (2009) have observed similar asymmetric risk in credit card markets.

Data Envelopment Analysis (DEA) models for classification problems were first used in the 1990s (Troutt, Rai, & Zhang, 1994). Currently, there are a lot of DEA models available for a variety of business analytics tasks, such as solving inverse classification problems (Pendharkar, 2002), data preprocessing (Pendharkar, 2005), fuzzy classification (Pendharkar, 2012), interactive classification (Pendharkar & Troutt, 2014), cluster analysis (Toloo, Saen, & Azadi, 2015), feature selection (Zhang et al., 2015), and dimensionality reduction (Pendharkar & Troutt, 2011). The DEA models can also be incorporated into radial basis neural networks to solving non-linearly separable classification problems that may contain inputs with negative values (Pendharkar, 2011a).

Most of the DEA based classification models use a dual variant of the variable returns-to-scale BCC model (Banker, Charnes, & Cooper, 1984) that provides non-linear (piecewise linear) classification decision boundaries. For two-class problems, a separate model is solved for each class, and two decision boundaries are obtained. When a problem is linearly inseparable, there are overlapping examples belonging to two different classes that appear between the two decision boundaries. However, when a classification problem is linearly separable, there are no such overlapping examples between the decision boundaries (see Fig. 1). For linearly separable problems, traditional margin maximizing support vector machines (SVMs) may be the best classifiers, and DEA models should not be used. DEA models are primarily suitable for linearly inseparable problems. These models may also be applied to non-linearly separable problems when combined with neural networks as hybrid models (Pendharkar, 2011a).

The primary contribution of this paper is to design and use DEA models for probabilistic classification. To our knowledge, probabilistic classification DEA models have never been used for such tasks. The existing DEA models for classification only predict binary class labels making them difficult to use for classification problems where misclassification cost is asymmetric. As mentioned before, there are three different kinds of risks associated with classification problems, and existing DEA classification models only address symmetric risk classification problems. Probabilistic classification DEA models are more general and can be used for both symmetric and asymmetric risk

E-mail address: pxp19@psu.edu.
URL: http://www.personal.psu.edu/pxp19/.

https://doi.org/10.1016/j.cie.2018.03.037
Received 24 September 2016; Received in revised form 6 February 2018; Accepted 22 March 2018
Available online 26 March 2018
0360-8352/

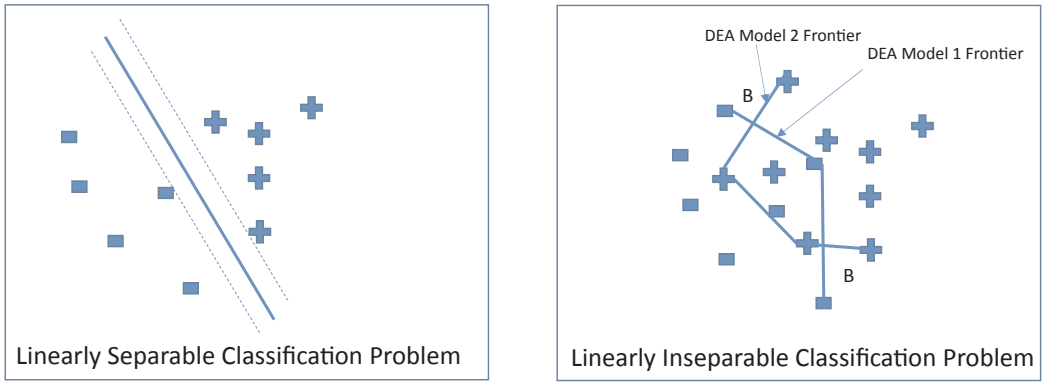

Fig. 1. Classification problem types and DEA model frontiers.

classification problems. These models can also allow decision-makers to specify subjective class prior probabilities to directly minimize Bayesian misclassification risk. This flexibility of allowing a decision-maker to provide subjective estimates to solve a broader range of classification problems (involving different kinds of risks) makes probabilistic DEA classification models more advanced than traditional DEA classification models.

There are two primary classification techniques to minimize Bayesian misclassification risk. The first one is an SVM that minimizes Bayesian misclassification cost risk. Pendharkar (2015) study used this technique and illustrated that these SVMs are very competitive and hard to outperform for linearly inseparable problems. The other technique available for minimizing Bayesian misclassification cost risk is the probabilistic neural network (PNN) classifier (Specht, 1990).

It is well known that DEA inefficiency scores may be modeled as exponential or half-normal distributions. Banker (1996) has often used these inefficiency score distributions for hypothesis testing. We believe that similar distributions may be used to develop probabilistic DEA classifiers. Additionally, a hybrid DEA-PNN model may also be used to further develop DEA based probabilistic classifiers. For the hybrid model, the DEA may be used for data preprocessing to select hard-to-classify region examples. These are then selected and used by the PNN model to learn class probability density functions (PDFs) using a Parzen window kernel estimation (Parzen, 1962).

In this paper, we develop three different probabilistic DEA based classifiers for linearly inseparable classification problems. The first two classifiers are built using exponential and half-normal DEA inefficiency score distribution assumptions. The third classifier uses DEA efficiency score threshold-based data preprocessing, to select high-efficiency examples to train a PNN for probabilistic estimation. Generally, selecting high-efficiency score examples that appear in the hard-to-classify region, and using these examples to train a PNN, may lead to improved learning of individual class PDFs (rather than using all examples, as is usually the practice in traditional PNN). We use simulated and real-world bankruptcy prediction datasets to test our approaches, and compare them with the cost-sensitive SVM and the traditional PNN that are known to minimize Bayesian misclassification cost risk. We use symmetric and asymmetric misclassification costs in our experiments.

This paper is organized as follows: In Section 2 we provide the classification problem definition and misclassification cost minimizing decision rule, review relevant DEA classification models from literature and propose minor extensions to one model. In Section 3 we propose two probabilistic DEA classification models for linearly inseparable classification problems and describe the competing Bayesian misclassification cost risk minimizing SVM and PNN models. In Section 4 we describe our data and experiments. In Section 5, we conclude our paper with a summary and discussion, and suggest a few directions for future research.

2. Problem statement, brief review of existing DEA classification models, and minor extensions

This section is divided into two parts. First, we provide the classification problem definition and Bayesian misclassification cost minimizing rule. We then review related DEA models for classification proposed in the literature and propose minor extensions to one of these models so that it becomes suitable for probabilistic classification.

2.1. Probabilistic classification problem and misclassification cost minimizing decision rule

For DEA based classification models, we consider a binary classification problem consisting of a vector $x_i = (x_{i1},\ldots,x_{im})^T \in \mathscr{R}^+$ of m attributes defined over a set of n input examples, where $i = 1,\ldots,n$. The primary objective of DEA based classification is to learn a function f: $(R^m)^+ \rightarrow \{0,1\}$, which best classifies the input examples. Our problem definition assumes that all components of the decision-making attribute vector are positive. Additionally, we assume that the classification function is probabilistic and minimizes misclassification cost.

Probabilistic classification requires the use of conditional probability density functions to identify the probability that an example x_i is a member of either class $c = 0$ or $c = 1$ (Duda & Hart, 1973). Assuming that $P(1)$ and $P(0)$ are class prior probabilities, and $P(x_i|0)$ and $P(x_i|1)$ are conditional probabilities that x_i is member of either class 0 or 1 respectively, then posterior probabilities, $P(c|x_i)$ for $c = 0,1$, are obtained using the Bayesian rule:

$$P(c|x_i) = \frac{P(x_i,c)}{P(x_i)}, \tag{1}$$

where

$$P(x_i,c) = P(x_i|c)P(c), \text{ and } P(x_i) = P(x_i,0) + P(x_i,1). \tag{2}$$

Since, for both values of $c = 0, 1$; the value of denominator in Eq. (1) remains same (see Eq. (2)), we have:

$$P(c|x_i) \propto P(x_i|c)P(c). \tag{3}$$

When probabilistic classification problems are used to minimize misclassification costs, we need to define additional misclassification cost variables. Assume that $C(1|0)$ represents the cost of misclassifying an example into class 1 when correct class assignment should be class 0, and $C(0|1)$ is defined similarly. The expected cost of assigning an example x_i to class $c = 0,1$ can be written as follows:

$$C_i(c) = C(c|0)P(0|x_i) + C(c|1)P(1|x_i). \tag{4}$$

The objective of misclassification cost minimizing classification problem is to determine a class $c \in \{0,1\}$ for each example x_i, so that following expression is minimized:

$$Minimize \sum_{i=1}^{n} C_i(c) \tag{5}$$

The optimal solution to the objective function can be found, for the training dataset, by using following Bayesian misclassification cost minimizing decision rule:

Decide class $c = 0$ for x_i if $C_i(0) < C_i(1)$; otherwise, decide class $c = 1$. (6)

Application of Bayesian misclassification cost minimizing decision rule for DEA classification problems requires additional DEA classification specific considerations that we describe later in Section 3.

2.2. Overview of DEA based classification and minor extensions

Before describing DEA models for classification, we introduce some preliminaries. First, we assume that all components of the decision-making attribute vector are positive. This assumption is necessary to introduce early DEA based classification models. These early models were later extended to relax this constraint. We also assume that the class labels for binary classification problem are 0 (reject) and 1 (accept). Additionally, we assume a set, Z, that contains indices of training examples that have a class label of 0. We also assume a set, O, that contains indices of training examples with class label 1. If the cardinality of sets is given by $|Z| = k$ and $|O| = l$, then $k + l = n$.

There are two different types of DEA classification models in the literature. Both of these models were largely developed independent of each other and were inspired by DEA additive and DEA ratio models. Both models appeared in literature nearly simultaneously. The DEA classification based on ratio model was proposed by Troutt et al. (1994), and the DEA classification based on the additive model was proposed by Sueyoshi (1999). We present both of these models in this section. We present DEA classification based on additive DEA model first. However, before we describe these models, we mention a few points about some differences between traditional DEA models and DEA models used for classification. The DEA models used for classification are slightly different from traditional DEA models that contain both input and output vectors. DEA classification models—either input-minimizing or output-maximizing—do not contain either an input or an output vector. The missing input or output is assumed to be a uni-dimensional vector, of constant value 1, and is omitted from DEA classification models. As a result, we do not use input or output vector terminology in this section. We note that non-traditional DEA models with constant input (Lovell & Pastor, 1999), missing inputs (Masoumzadeh, Toloo, & Amirteimoori, 2016; Toloo, 2013) or missing outputs (Toloo & Kresta, 2014) are well studied in DEA literature. A recent study (Toloo & Tavana, 2017) investigates some of the computational challenges associated with these models, and proposes a computationally efficient procedure to solve them.

We also assume that the classification problem is linearly inseparable (i.e. class overlap is present). A reader may make a few observations. First, linearly separable classification problems are simple problems and support vector machines may be used for maximum margin classification. Such problems are not suitable for the application of DEA based classifiers (see Fig. 1 for linearly separable and inseparable problems for two attribute datasets). Second, non-linearly separable problems may be converted into linearly separable, or inseparable problems, using Kernel transformation. Third, multi-class classification problems may be converted into a series of binary classification problems using error correcting output codes (Witten, Frank, & Hall, 2011).

The additive DEA classification model (Sueyoshi, 1999) requires two steps to learn a classification function. In the first step, following goal programming formulation is solved to identify classification overlap:

$$\begin{aligned} Minimize \sum_{i\in O} \chi_{1i}^{+} + \sum_{i\in Z} \chi_{0i}^{-}, \text{subject to: } & \sum_{j=1}^{m} \beta_j x_{ij} + \chi_{1i}^{+} - \chi_{1i}^{-} = d,\ i \in O, \\ & \sum_{j=1}^{m} \gamma_j x_{ij} + \chi_{0i}^{+} - \chi_{0i}^{-} = d-\eta,\ i \in Z. \\ & \sum_{j=1}^{m} \beta_j = 1, \sum_{j=1}^{m} \gamma_j = 1, \text{all slacks} \geqslant 0,\ \beta_j \geqslant 0, \gamma_j \geqslant 0 \text{ and } d\text{:unrestricted}, \end{aligned} \tag{7}$$

where χ_{1i}^{+} and χ_{1i}^{-} are positive and negative deviations from the piecewise linear frontier $\sum_{j=1}^{m} \beta_j x_{ij} = d$. The constant $\eta > 0$ is necessary to avoid trivial all zero solution and is usually a very small number. The slacks χ_{1i}^{+} and χ_{1i}^{-} are positive and negative deviations from the piecewise linear frontier $\sum_{j=1}^{m} \gamma_j x_{ij} = d-\eta$. Given a new test case x^t, its class can be identified using following criteria:

(a) If $\sum_{j=1}^{m} \beta_j^* x_j^t \geqslant d^*$ and $\sum_{j=1}^{m} \gamma_j^* x_j^t \geqslant d^*$ then Accept the case.
(b) If $\sum_{j=1}^{m} \beta_j^* x_j^t < d^*$ and $\sum_{j=1}^{m} \gamma_j^* x_j^t < d^*$ then Reject the case.
(c) If $\sum_{j=1}^{m} \beta_j^* x_j^t > d^* \geqslant \sum_{j=1}^{m} \gamma_j^* x_j^t$ or $\sum_{j=1}^{m} \beta_j^* x_j^t \leqslant d^* < \sum_{j=1}^{m} \gamma_j^* x_j^t$ then case belongs to overlapping region.

For cases belonging to the overlapping region, following second stage DEA classification problem was solved to classify them into one of the two classes:

$$\begin{aligned} Minimize \sum_{i\in O} \chi_{1i}^{+} + \sum_{i\in Z} \chi_{0i}^{-}, \text{subject to: } & \sum_{j=1}^{m} \pi_j x_{ij} + \chi_{1i}^{+} - \chi_{1i}^{-} = d,\ i \in O, \\ & \sum_{j=1}^{m} \pi_j x_{ij} + \chi_{0i}^{+} - \chi_{0i}^{-} = d-\eta,\ i \in Z. \\ & \sum_{j=1}^{m} \pi_j = 1, \text{all slacks} \geqslant 0,\ \pi_j \geqslant 0 \text{ and } d\text{: unrestricted}, \end{aligned} \tag{8}$$

The overlapping region cases can be either classified into class 1 (accept) or class 0 (reject) based on whether these cases lie on or above the discriminant line $\sum_{j=1}^{m} \pi_j^* x_{ij} = d^*$ or below it. Cases falling below the line were classified into class 0, otherwise these were classified as those belonging to class 1 (Sueyoshi, 1999). This original additive DEA classification model was later extended to incorporate negative variables (Sueyoshi, 2001), and, in another study, a computationally intensive mixed integer programming formulation was proposed to resolve cases belonging to class overlap region (Sueyoshi, 2004).

Toloo et al. (2015) illustrated the limitations of model (7) in that the results of this model were sensitive to the value of parameter $\eta > 0$. They illustrate that the model (7) provides different results for different values of η, and for higher values of η, the model even fails to solve the most trivial discrimination problem with two groups that are linearly separable with no group classification overlap. To resolve this limitation, Toloo et al. (2015) proposed following formulation to identify group classification overlap:

$$\begin{aligned} Maximize\ \eta, \text{subject to: } & \sum_{j=1}^{m} \beta_j x_{ij} - \chi_{1i}^{-} = d, i \in O, \sum_{j=1}^{m} \gamma_j x_{ij} + \chi_{0i}^{+} = d-\eta, i \in Z. \\ & \sum_{j=1}^{m} \beta_j = 1, \sum_{j=1}^{m} \gamma_j = 1, \text{all slacks} \geqslant 0,\ \beta_j \geqslant 0, \gamma_j \geqslant 0;\ d \text{ and } \eta\text{:unrestricted}. \end{aligned} \tag{9}$$

Model in formulation (9) removes arbitrariness of η from model in formulation (7) and is slightly simple because it involves fewer variables.

In case of classification problems that are linearly inseparable, group overlap exists. Toloo et al. (2015) argued that the second stage

classification overlap resolution formulation (8) is also flawed due to its arbitrary value of constant $\eta > 0$. To resolve this issue, Toloo et al. (2015) proposed following second-stage formulation for resolving classification cases belonging to the group overlapping region:

$$Maximize\ \eta, \text{subject to: } \sum_{j=1}^{m} \pi_j x_{ij} - \chi_{1i}^{-} = d,\ i \in O, \sum_{j=1}^{m} \pi_j x_{ij} + \chi_{0i}^{+} = d - \eta,\ i \in Z. \sum_{j=1}^{m} \pi_j = 1, \text{all slacks} \leqslant 0,\ \pi_j \geqslant 0;\ \eta \text{ and } d\text{: unrestricted.} \quad (10)$$

The ratio DEA classification model was proposed by Troutt et al. (1994). It provides one non-linear (piecewise linear) classification boundary with datasets containing examples from only one class. Troutt et al. (1994) model assumes that examples from only one class (accept class) are available, and learning an efficient frontier for the accepted cases represented the classification frontier. Assuming that the training dataset contains k examples from accept class, this efficiency frontier was learned by solving the following DEA efficiency program for each of these $i = \{1,\ldots,k\}$ examples:

$$\max \phi_i = \frac{a_1}{\sum_{j=1}^{m} b_j x_{ij}}, \text{subject to: } \frac{a_1}{\sum_{j=1}^{m} b_j x_{ij}} \leqslant 1, \text{ for all } i = 1,\ldots,k; a_1, b_j \geqslant 0. \quad (11)$$

All examples with $\phi_i^* = 1$ were deemed efficient. Assuming that E^* is the set of all efficient cases in training data then given a new test case x^t, following DEA efficiency formulation was solved to determine its class prediction:

$$\max \phi_i = \frac{a_1}{\sum_{j=1}^{m} b_j x_{ij}^t}, \text{Subject to: } \frac{a_1}{\sum_{j=1}^{m} b_j x_{ij}} \leqslant 1 \text{ for all } i \in E^* \cup \{x^t\}; a_1, b_j \geqslant 0. \quad (12)$$

Assuming a_1^* and b_j^* as optimal weights in formulation (12), following rule was used to classify the new test case:

$$\text{If } \frac{a_1^*}{\sum_{j=1}^{m} b_j^* x_{ij}^t} < 1 \text{ Then accept the case.}$$
$$\text{Else if } x^t \text{ is efficient and does not alter } E^*$$
$$\text{Then accept the case, otherwise reject the case.} \quad (13)$$

Pendharkar (2011a) further extended DEA ratio model to linearly inseparable classification problems containing examples from both classes. In our research, we use and extend the Pendharkar (2011a) model for probabilistic classification, which is also known as the DEA model in envelopment form (Lovell & Pastor, 1999). However, before we explain our extensions, we first describe the Pendharkar (2011a) model. While this model assumes that the class labels for binary classification problem are 0 and 1, these class labels are not determined arbitrarily. Assuming that μ^0 and μ^1 are mean vectors of m-attributes that belong to the training dataset, then class labels should be such that $\|\mu^1\| > \|\mu^0\|$. If a violation occurs, class relabeling must be carried out for both training and holdout datasets so that this condition is satisfied. Two separate frontiers are developed separately for each class using two different DEA models (Pendharkar, 2011a).

Fig. 1 shows the two frontiers for a linearly inseparable classification problem. Model 1 frontier is drawn using class 0 examples from the training dataset (i.e. $i \in Z$). We assume that all the examples belonging to class 0 are now labeled vector $x_z = (x_{z1},\ldots,x_{zm})$ with $z = 1,\ldots,k$. For each of these examples $i = \{1,\ldots,n\}$ $i \in Z$, the efficiency, $\xi_i = 1/\phi_i^*$, is computed by solving the following output maximizing DEA, which we title as Model 1 (Pendharkar, 2011a).

$$Max \phi_i, \text{Subject to: } \sum_{z=1}^{k} \lambda_z x_{zj} - \phi_i x_{ij} \geqslant 0, j = 1,\ldots,m; \sum_{z=1}^{k} \lambda_z = 1, \phi_i, \lambda_z \geqslant 0, z = 1,\ldots,k. \quad (14)$$

Similarly, we assume that all the examples belonging to class 1 are labeled as $x_u = (x_{u1},\ldots,x_{um})$ with $u = 1,\ldots,l$. For each of these examples $i = \{1,\ldots,n\}$ and $i \in O$, the efficiency, $\xi_i = \phi_i^*$, is computed by solving the following input minimizing DEA, which we title Model 2 (Pendharkar, 2011a).

$$Min \phi_i, \text{Subject to: } \sum_{u=1}^{l} \lambda_u x_{uj} - \phi_i x_{ij} \leqslant 0, j = 1,\ldots,m; \sum_{u=1}^{l} \lambda_u = 1, \phi_i, \lambda_u \geqslant 0,\ u = 1,\ldots,l. \quad (15)$$

Pendharkar (2011a) did not specify any procedure to identify when a test example may be considered to fall under overlap region, but he provided a nearest neighborhood classification rule to classify a test example belonging to the overlap region by taking projections of this test example on Model 1 and Model 2 efficiency frontiers. If for x^t test example, x^{t1} and x^{t2} are its Model 1 frontier and Model 2 frontier projections then the Euclidean distances of the test example from its projections were used to classify the test example as either belonging to class 1 or class 0. Specifically, following rule was used to classify test example belonging to class overlap region: If $\|x^t - x^{t1}\| > \|x^t - x^{t2}\|$ then class = 1, elseif $\|x^t - x^{t1}\| < \|x^t - x^{t2}\|$ then class = 0; else class = random {0,1}, where random {0,1} is random classification of an example to either class, using uniform class assignment probability.

In our minor extension to Pendharkar (2011a) model, we provide a simple procedure to include negative decision-making attributes and provide rules (Table 1) to identify the location of test example with respect to two efficiency frontiers. Additionally, we note the existence of "blind" classification regions in Pendharkar (2011a) model and highlight that the proposed rules are useful in identifying whether classification overlap region exists. Formal identification of classification overlap region is important because our proposed probabilistic DEA classification model should not be used for classification problems that do not contain classification overlap region.

Negative decision-making attributes can be incorporated into DEA efficiency analysis. For example, semi-oriented radial measure models allow a decision-maker to incorporate negative attributes (Emrouznejad & Latef, 2010) in DEA models. Additionally, neural network models can also be used for efficiency analysis of large-scale data containing negative attributes (Toloo, Zandi, & Emrouznejad, 2015). In our extension, when decision-making attributes are negative, we consider two possible transformations to make these decision-making attributes positive. The first approach uses a positive definite Kernel transformation similar to the one used in Kernel-based SVMs. The second transformation is to add a positive constant $K = abs(\min_{i=1,\ldots,n, j=1,\ldots m}(x_{ij})) + \psi$ to each attribute vector of the entire training and test datasets. The second transformation moves the entire dataset into the positive quadrant, and the mean value of each attribute is increased by K; the standard deviation remains unchanged. As an example, the following matrix D illustrates three input vectors in a dataset $= \begin{bmatrix} 3 & -10 & 5 \\ 4 & -3 & 10 \\ 5 & 3 & 35 \end{bmatrix}$, then $abs(\min_{i=1,\ldots,n, j=1,\ldots m}(x_{ij})) = 10$. Assuming some arbitrary positive constant $\psi = 5$, we get $K = 15$. Adding this value of K to each attribute, the

Table 1
Test vector location based on its efficiency score.

Model 1 efficiency score	Model 2 efficiency score	Region
Less than 1	Less than 1	Overlap
Less than 1	1	Under Model 1 frontier
1	Less than 1	Above Model 2 frontier
1	1	Blind

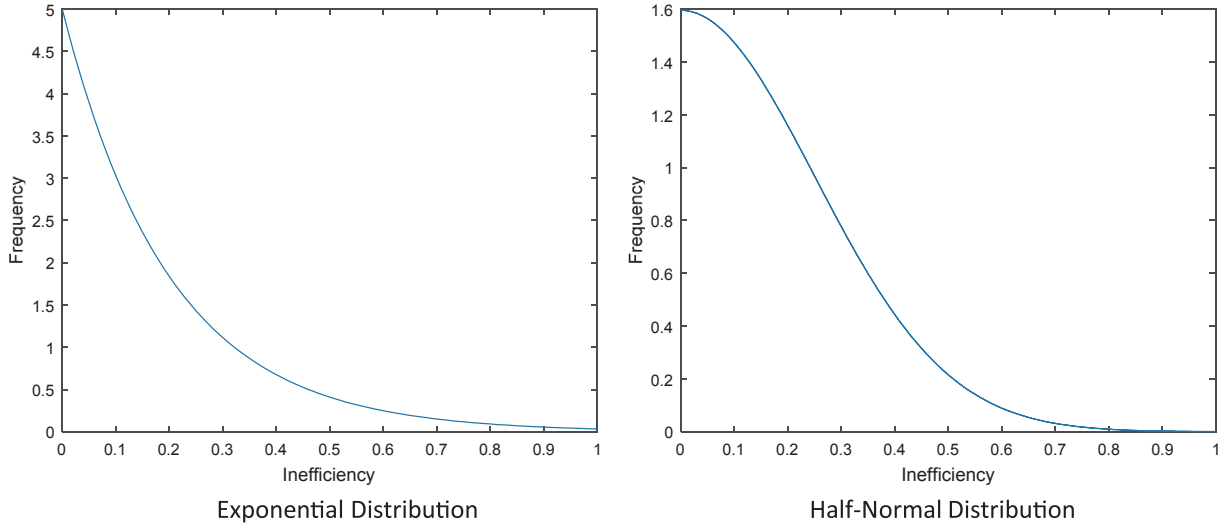

Fig. 2. Inefficiency score distributions.

transformed dataset D' is $D' = \begin{bmatrix} 18 & 5 & 20 \\ 19 & 12 & 25 \\ 20 & 18 & 50 \end{bmatrix}$. Care must be taken in selecting a value of $\psi > 0$ to ensure that when K is added (to training and future test or currently available holdout samples), the classification problem always remains in the positive quadrant. Thus, ψ may be a small or large positive constant. In our research, for datasets with negative attributes, we use the second transformation. The reason for this is its simplicity, and it allows us to compare our techniques with a SVM that uses a linear Kernel. For our purpose, we relegate the use of non-linear Kernel transformations to non-linearly separable classification problems.

To identify the location of a training data example with respect to two frontiers, two important observations may be made. First, the efficiency score, ξ_i, of each example, may be used to indicate the distance of that example from the frontier. Lower efficiency scores indicate that an example is far from the frontier and is easy to classify. Higher efficiency scores indicate that the example is either on the frontier (when $\xi_i = 1$), or close to it. In both cases, the example falls into the "difficult-to-classify" region because the classification problem is linearly inseparable. Second, examples lying inside the overlap region need to be clearly identified. To do this, we assume that R^0 and R^1 are DEA reference sets (i.e. examples that get values of $\xi_i = 1$) for Model 1 and Model 2. To identify examples that fall inside the overlapping region additional DEA analysis is required. First, we conduct DEA analysis using Eq. (14) and reference set R^0 for each of the examples $i \in O$ one at a time. The examples that get a value of $\xi_i < 1$ with $i \in O$ fall below the Model 1 frontier (overlapping region). Similarly, the second DEA analysis uses Eq. (15) and reference set R^1 for each of individual examples in set Z, where $i \in Z$. The examples that get a value of $\xi_i < 1$ with $i \in Z$ fall above the Model 2 frontier (overlapping region). This analysis will also indicate if a problem is linearly separable, because in such cases, no examples will be found in the overlapping region. For linearly separable classification problems, SVM models may be better suited for solving the classification problem and DEA models should not be used any further.

The location of a test example vector x^t (also called the holdout sample example), with respect to DEA frontiers, may be identified by solving two similar DEA models. For Model 1, the efficiency of the test vector is computed using dataset $x^t \cup R^0$ and Eq. (14). For Model 2, the efficiency for the test vector is computed using dataset $x^t \cup R^1$ and Eq. (15). There are four regions a test vector may belong to: below Model 1 frontier (outside the overlap area), above Model 2 frontier (outside the overlap area), inside overlap area, or in one of the blind regions shown using the letter "B" in Fig. 1. Table 1 illustrates how the efficiency score of the test example from two models may be used to identify its location.

3. Probabilistic DEA model and Bayesian misclassification cost minimizing benchmarking classification models

This section is divided into three sub-sections. First, we describe class PDF estimation using exponential and half-normal inefficiency PDF assumptions. These probability estimations are foundations for probabilistic DEA classification for exponential and half-normal inefficiency PDFs. In our second sub-section, we provide an overview of traditional PNN and hybrid DEA-PNN. Both of these techniques directly use the Bayesian misclassification cost minimizing classification rule. Finally, we describe one of the best-known Bayesian misclassification cost minimizing SVMs. This SVM and traditional PNN serve as benchmarks to evaluate the three different probabilistic DEA models proposed in our study.

3.1. DEA classification probability estimation and classification rule

The procedure to estimate class probabilities is carried out for holdout/test sample examples only. The training dataset is used only to provide reference sets R^0 and R^1. These reference sets are then used for each example in the holdout sample as described at the end of Section 2. Let ξ^{t0} and ξ^{t1} be the Model 1 and Model 2 efficiency scores for a test example $t = 1,\ldots,s$. Corresponding to each efficiency score, and for the purpose of this research, we define variables ε^{t0} and ε^{t1}, called the inefficiency scores[1] for test example t. The value of this variable can be obtained using the following expression:

$$\varepsilon^{tc} = 1 - \xi^{tc}, \text{ where } t = 1,\ldots,s, \text{ and } c = 0,1. \quad (16)$$

Like the efficiency scores, the inefficiency scores can take values between zero and one. The DEA literature commonly uses exponential and half-normal distributions for inefficiency score distributions (Banker, 1993). Fig. 2 illustrates these two distributions. Banker et al. (1990) argue that the exponential distribution is appropriate when an analyst believes that a lot of observations will lie close to a DEA frontier; a half-normal distribution is appropriate when an analyst expects that fewer observations will lie close to DEA frontier.

[1] This is also referred to as efficiency score deviations in literature (Banker, Kauffman, & Morey, 1990).

In our research, we assume that both training and test examples are drawn from the same underlying population distribution. We call this assumption *representativeness*. It is similar to the common production process assumption in DEA literature. Depending on the inefficiency score distribution assumption of the analyst, the parameters of distributions will be learned using the inefficiency scores of training dataset examples, and a Maximum Likelihood estimation procedure. Let ε_i represent the inefficiency score for each of ith training examples. Assuming exponential distributions for class 0 and class 1, the PDFs for class 0 (i.e., $i \in Z$) and class 1 ($i \in O$) inefficiency scores may be represented as follows:

$$f(\varepsilon^0) \sim exp(\alpha^0) \quad \text{for } i \in Z, \text{ and } f(\varepsilon^1) \sim exp(\alpha^1) \quad \text{for } i \in O. \tag{17}$$

The parameters of the distributions, α^0 and α^1, are estimated using the following expressions:

$$\alpha^0 = \frac{1}{k}\sum_{i=1,i\in Z}^{n} \varepsilon_i, \text{ and } \alpha^1 = \frac{1}{l}\sum_{i=1,i\in O}^{n} \varepsilon_i \text{ respectively.} \tag{18}$$

The representativeness assumption assumes that the parameters computed using a training data distribution will be similar to inefficiency distributions that may be seen in holdout samples.

When a half-normal distribution assumption is made for class 0 and class 1 inefficiency distribution scores, the PDFs for these inefficiency scores may be represented as follows:

$$f(\varepsilon^0) \sim HN(\sigma^0) \quad \text{for } i \in Z, \text{ and } f(\varepsilon^1) \sim HN(\sigma^1) \quad \text{for } i \in O. \tag{19}$$

The parameters of distributions, σ^0 and σ^1, are estimated using following expressions:

$$\sigma^0 = \sqrt{\frac{1}{k}\sum_{i=1,i\in Z}^{n} \varepsilon_i^2}, \text{ and } \sigma^1 = \sqrt{\frac{1}{l}\sum_{i=1,i\in O}^{n} \varepsilon_i^2}. \tag{20}$$

While it is certainly possible to consider mixed inefficiency score distributions for class 0 and class 1 training dataset examples (e.g., $f(\varepsilon^0) \sim exp(\alpha^0)$ and $f(\varepsilon^1) \sim HN(\sigma^1)$), we do not consider such distributions here, to keep the scope of our research more focused.

For a test example x^t, with its inefficiency scores ε^{t0} and ε^{t1} and exponential inefficiency score distributions, the probabilities that the example belongs to class 0 (i.e., $P(0|x^t)$) and class 1 can be estimated using a cumulative distribution function (CDF) of the exponential distribution as follows:

$$P(0|x^t) \propto 1-e^{-\alpha^0\varepsilon^{t0}} \text{ and } P(1|x^t) \propto 1-e^{-\alpha^1\varepsilon^{t1}}. \tag{21}$$

A reader may note that these probabilities are not normalized, and indeed normalization is not necessary to apply the Bayesian classification rule that we will describe below.

For half-normal inefficiency score distributions, similar probabilities may be computed using CDF for the half-normal distribution as follows:

$$P(0|x^t) \propto \Phi\left(\frac{\varepsilon^{t0}}{\sigma^0\sqrt{2}}\right), \text{ and } P(1|x^t) \propto \Phi\left(\frac{\varepsilon^{t1}}{\sigma^1\sqrt{2}}\right). \tag{22}$$

The $\Phi(v)$ in Eq. (22) is an error function defined as:

$$\Phi(v) = \frac{2}{\sqrt{\pi}}\int_0^v e^{-q^2}dq. \tag{23}$$

In our research, we approximate the value of the error function using its Maclaurin series as:

$$\Phi(v) \approx \frac{2}{\sqrt{\pi}}\left(v - \frac{v^3}{3} + \frac{v^5}{10}\right) \tag{24}$$

Since the test example is classified using the Bayesian classification rule, we describe additional DEA classification specific considerations needed to apply this classification rule. Assuming that the misclassification cost is zero for correct classification, i.e. $C(1|1) = C(0|0) = 0$, the Bayesian misclassification cost minimizing classification rule picks a class c based on the following rule:

$$\text{IF } P(1|x^t) \times C(0|1) > P(0|x^t) \times C(1|0) \text{ THEN } c = 1, \text{ ELSE } c = 0. \tag{25}$$

The rule in Eq. (25) will classify most holdout sample cases. An exception will occur when the test example x^t falls into the blind region "B" shown in Fig. 1. In such an event both $P(1|x^t) = P(0|x^t) = 0$. When this situation occurs, special consideration is necessary. If misclassification costs are asymmetric ($C(0|1) \neq C(1|0)$), then the example is classified into a class that yields lower misclassification cost. This process can be done by changing $P(1|x^t) = P(0|x^t) = 1$ and then applying the rule (Eq. (25)). However, when misclassification costs are symmetric ($C(0|1) = C(1|0)$), then the example is classified in a class that has higher prior probability. If prior probabilities are equal then the decision maker needs to provide a preferred default class.

The representativeness assumption does not require a decision-maker to provide prior probabilities because these can be determined using examples from the training dataset. In our research, we do this, and we assume that the preferred class is class 0. However, we do recognize different possibilities, where prior probabilities may be supplied by a decision-maker, or by a combination of decision-maker supplied prior probabilities that may be updated using observed training data.

3.2. Hybrid DEA probabilistic neural network

PNN is a pattern classifier that uses a Parzen density estimation (Parzen, 1962) and Bayesian decision rule for classification (Specht, 1990). A typical PNN with binary outputs is illustrated in Fig. 3. In this figure, the number of input nodes is equal to the number of decision-making attributes, and the number of pattern nodes is equal to the number of examples in the training data set. A PNN stores all training examples in memory. For a test example t, the input nodes will be assigned values of m corresponding to different attributes of x^t. For each of the $i = \{1,..,n\}$ pattern nodes, the value of its output o_i can be computed as $o_i = e^{-\sum_{j=1}^{m}\frac{(x_{ij}-x_j^t)^2}{2\sigma^2}}$, where σ is a smoothing parameter for the Gaussian kernel. The third layer contains summation nodes, which sum the outputs o_i of patterns belonging to each class in training data. Using the previously described convention (set Z containing indices of training examples that have a class label of 0, and set O that with indices of class label 1), the summation of outputs for class 0 (S_0) and class 1 (S_1) are computed as follows: $S_0 = \frac{1}{k}\sum_{i\in Z} o_i$ and $S_1 = \frac{1}{l}\sum_{i\in O} o_i$. The last layer is the output layer, which classifies summation layer outputs to one of the two classes using prior probabilities for each class, and applying the Bayesian decision rule. If $P(c)$ denotes the prior

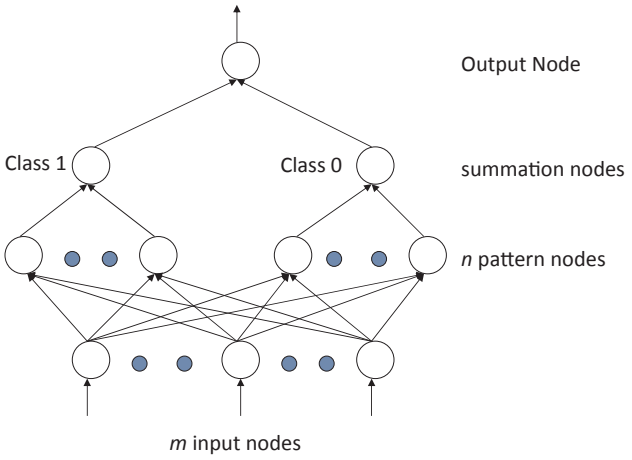

Fig. 3. A probabilistic neural network for classification.

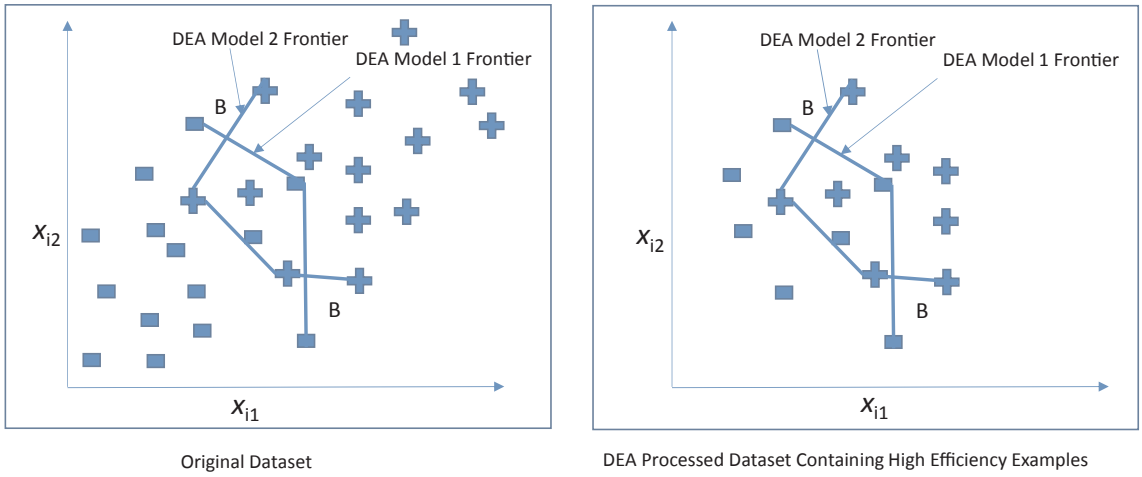

Fig. 4. Original training dataset vs. DEA processed dataset.

probability element $c \in \{0,1\}$ then the output node, for each example, chooses a class c that maximizes the expression $\underset{c\in\{0,1\}}{\text{argmax}}[P(c) \times S_c]$. The key design parameter in the performance of PNN is selection of an appropriate value of σ. We optimize this parameter using a bisection procedure described in Pendharkar (2011b).

A hybrid DEA-PNN will use DEA Models 1 and 2 to select certain sub-sets of training dataset examples for learning PNN. That is, instead of using all n examples in a PNN, a hybrid DEA-PNN will consider fewer training dataset examples that have high ξ_i values. As mentioned before, examples with low ξ_i values are farther away from the frontiers and are easy to classify. Eliminating these examples may improve the predictive capabilities of a PNN because the class PDF distribution will be learned using difficult to classify examples. Fig. 4 illustrates a two-attribute example of the original training dataset, and a DEA processed training sub-set with high ξ_i value examples. There are fallacies in DEA data processing as well. It may be argued that DEA data processing may select too few examples for the training dataset. This can happen when there are only a few examples in the overlapping zone and original training dataset is small. Rejecting low-efficiency examples from smaller original training datasets may hurt the learning of class PDFs. While there may well be such clinical cases, in general, the DEA is well known to be biased in assigning high-efficiency scores to examples (Dyson & Thannassoulis, 1988). For large datasets, with the confirmed existence of overlapping regions, DEA data processing should usually help. Additionally, even for smaller datasets, attrition of case selection via DEA data processing may be controlled by using lower threshold efficiency scores to reject cases from the original training dataset. For our research, we use a threshold of 0.9. If an example in the original training dataset has $\xi_i < 0.9$, then that example is eliminated and not included in the DEA processed training dataset. For the sake of comparison, we use both traditional PNN on the original training dataset, and another hybrid DEA-PNN on the DEA processed training dataset. By using both of these PNNs, it is easy to identify if there are any performance improvements via DEA data processing.

When misclassification costs are considered, the Bayesian misclassification cost risk minimizing rule from Eq. (25) is applied for classification. The details are as follows. From Bayesian theory, we have $P(x^t|c) \propto S_c$. Furthermore, from Bayes Theorem, $P(c|x^t) \propto P(x^t|c) \times P(c)$. By using $P(c|x^t) = P(x^t|c) \times P(c)$, we can apply Eq. (25) to classify a test example. Prior probabilities are computed using a number of examples belonging to each class in the original training dataset. The representativeness assumption made earlier assumes that prior probabilities of classes observed in the real world are represented in the original training dataset. When any violations of these assumptions do occur, then these priors may be provided by an expert.

3.3. Bayesian misclassification cost minimizing support vector machine

We use a Bayesian misclassification cost minimizing SVM from Pendharkar (2015) study. Assuming an m-dimensional vector $\beta = (\beta_1, \ldots,\beta_m)^T \in \Re$, and $C(0|1) \geq (2 \times C(1|0) - 1)$, this SVM can be mathematically formulated as follows:

$$\text{Minimize} \left\{\frac{1}{2}||\beta||^2 + \Delta\left[C(0|1)\sum_{i\in O}\delta_i + (2C(1|0)-1)\sum_{i\in Z}\delta_i\right]\right\} \text{Subject to}$$
$$: \beta^T x_i + \beta_0 \geqslant 1-\delta_i\ ,\ \forall\ i \in O, \beta^T x_i + \beta_0 \leqslant \delta_i - \theta\ ,\ \forall\ i \in Z, \theta = \frac{1}{(2C(1|0)-1)}, \beta_0, \beta_j \text{ unrestricted};\ \delta_i > 0\ ,\ \forall\ i \in \{1,\ldots,n\},\ j = \{1,\ldots,m\}. \quad (26)$$

The formulation (26) assumes that $C(0|1) \geq (2 \times C(1|0) - 1)$. In the event where $C(0|1) < (2 \times C(1|0) - 1)$, then the training and holdout sample data can be relabeled by swapping labels and costs $C(0|1)$ and $C(1|0)$, so that the new data adheres to the original assumptions. If classes are relabeled, then they are relabeled for both training and test datasets to ensure consistency. For a test example, if $\beta^T x^t + \beta_0 \geqslant 0$ it is classified into class 1, otherwise it is classified into class 0. We solve the formulation (26) in the primal using CPLEX software. We optimize the value of Δ from a set of values $\{2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3\}$ so that the resulting value minimizes training data misclassification cost, and not necessarily the objective function of the formulation. Thus, for a given training dataset, we run procedure (26) seven times (once for each value of Δ), and use the model where Δ provides the lowest misclassification cost for the training dataset. Pendharkar (2015) compared formulation (26) to several cost-sensitive linear classifiers, and found that the SVM is very robust, and performs very well against other competing linear classifiers.

4. Data, experiments, and results

We perform our experiments using simulated datasets and real-world bankruptcy prediction datasets. All datasets have continuous decision-making attributes associated with two classes, and the classification problem is linearly inseparable. Simulated data allows us to create ideal conditions where the representativeness assumption is satisfied, although violations of this assumption may occur in real-world datasets. They also allow us to simulate parametric normal and non-normal class PDFs. The real-world datasets contain non-parametric PDFs as they would normally occur in real-world decision-making situations. By testing our techniques across different datasets, we can

Table 2
Misclassification cost statistics for zero misclassification cost asymmetry.

Distribution	DEA-PNN	EDEA	HNDEA	PNN	SVM
Exponential	22.20	54.40	59.20	27.25	30.50
	(4.86)	(9.24)	(9.90)	(4.89)	(4.58)
Normal	43.75	43.90	44.00	69.45	39.50
	(6.67)	(5.95)	(6.12)	(7.21)	(5.56)
Uniform	39.50	48.85	48.90	49.65	44.20
	(4.90)	(5.89)	(6.01)	(7.21)	(4.09)

Table 3
Misclassification cost statistics for low misclassification cost asymmetry.

Distribution	DEA-PNN	EDEA	HNDEA	PNN	SVM
Exponential	33.35	80.05	88.05	36.85	51.65
	(6.58)	(18.66)	(20.45)	(7.15)	(6.38)
Normal	62.90	59.55	59.15	92.30	55.15
	(9.54)	(9.02)	(8.67)	(11.50)	(7.46)
Uniform	53.85	59.60	60.15	70.75	62.10
	(6.38)	(6.02)	(6.38)	(10.72)	(9.22)

Table 4
Misclassification cost statistics for high misclassification cost asymmetry.

Distribution	DEA-PNN	EDEA	HNDEA	PNN	SVM
Exponential	51.75	132.75	140.95	54.30	84.30
	(7.06)	(37.5)	(40.2)	(13.16)	(8.21)
Normal	83.85	81.35	80.90	127.40	73.45
	(12.04)	(13.55)	(12.79)	(18.2)	(11.96)
Uniform	70.90	78.05	78.95	94.35	77.90
	(6.96)	(8.61)	(9.55)	(14.36)	(10.04)

observe the robustness of each one.

Our simulated datasets were generated using SAS software procedures, for five decision-making attributes associated with two different classes, and three different group distributions each taking a value from the set D = {Normal, Exponential, Uniform}. Each dataset contained 300 examples, with 150 examples belonging to group 1 and the rest belonging to group 0. The group means for each of the decision-making attributes were set to 100 for group 1 and 90 for group 0. The standard deviation for all decision-making attributes was assigned a constant value of 10. A total of five datasets were generated for each of three different group distributions. For five datasets and each data distribution, $P(5,2) = 20$ unique permutations of training and holdout samples were used for our experiments. In all experiments with simulated datasets, three different misclassification cost asymmetries were defined so that each of these asymmetries took a unique value from the set $ = {Zero, Low, High}. In the case of Zero information asymmetry, all misclassification costs were assumed to be equal to one, and all correct classification costs were assumed to be equal to zero. For Low asymmetry the misclassification cost category was assigned the following values: $C(1|1) = C(0|0) = 0$, and $C(1|0) = 2$, $C(0|1) = 1$. A High asymmetric misclassification cost category was assigned the values: $C(1|1) = C(0|0) = 0$, and $C(1|0) = 4$, $C(0|1) = 1$.

Tables 2–4 illustrates holdout sample results for different misclassification cost asymmetries. In the results presented, EDEA refers to exponential DEA and HNDEA refers to half-normal DEA. Results in each cell are averaged for 20 holdout sample tests for each of the different techniques and data distributions. The results indicate a pattern. For a normal data distribution, cost-sensitive SVM provides a lower misclassification cost average; for exponential and uniform data distributions, hybrid DEA-PNN provides the lower average. The PNN appears to consistently provide the worst average for normal data distributions, and the hybrid DEA-PNN exhibits this trend also.

We perform the analysis of variance (ANOVA) statistical test with misclassification cost as a dependent variable, and distribution (D), technique (T) and misclassification cost ($) as independent factors. Table 5 illustrates the ANOVA results. They indicate that for all three factors, their two-way and three-way interactions were significant at 99% level of statistical confidence. The R-squared value was about 81%.

Table 5
The full factor ANOVA summary table.

Source	Type III sum of squares	Degrees of freedom	Mean square	F-ratio	Significance
Model	591521.7	44	13443.7	85.64	0.000**
Intercept	3742548.2	1	3742548.2	23841.77	0.000**
T	65143.7	4	16285.9	103.75	0.000**
D	4928.9	2	2464.5	15.70	0.000**
$	281624.6	2	140812.3	897.04	0.000**
T × D	194212.9	8	24276.6	154.65	0.000**
T × $	5840.3	8	725.5	4.62	0.000**
D × $	11859.9	4	2964.9	18.89	0.000**
T × D × $	27947.2	16	1746.70	11.13	0.000**
Error	134213.1	855	156.9		
Total	4468283.0	900			
Corr. total	725734.8	899			

R-squared = 0.815 (adjusted R-squared = 0.806); **Significant at 99% level of statistical significance.

Table 6
Overall pairwise difference in means Tukey's post hoc test for technique factor.

Technique	Subset 1	Subset 2	Subset 3	Subset 4
DEA-PNN	51.34			
SVM		57.64		
PNN			69.14	
EDEA			70.94	70.94
HNDEA				73.36

Mean square error = 156.974; Harmonic mean sample size = 180.

Table 7
Overall pairwise difference in means Tukey's post hoc test for distribution factor.

Distribution	Subset 1	Subset 2
Uniform	62.51	
Exponential	63.17	
Normal		67.77

Mean square error = 156.974; Harmonic mean sample size = 300.

Given the statistical significance of all three factors in predicting misclassification cost, we performed a post hoc analysis for the pairwise difference in means using Tukey's method, and represent the results in homogeneous subsets at 95% level of statistical confidence. Table 6 illustrates the results for the technique factor. Techniques with no difference in means belong to the same subset. The results indicate that, overall, the DEA-PNN outperformed all other techniques, followed by the SVM. No difference in means was observed for PNN and EDEA; and for EDEA and HNDEA techniques.

Tukey's homogeneous subsets for distribution and cost factors are shown in Tables 7 and 8. The results indicate no difference in means for Uniform and Exponential distributions. The mean values in Tables 2–4 indicate that the DEA-PNN had the lowest means for uniform and exponential distributions. However, for normal distributions, SVM had the lowest means.

As a hybrid technique, the DEA-PNN selects a subset of training examples to learn class distribution PDFs, and PNN uses all of the training examples. Of 300 examples in the training dataset, the DEA-PNN used about 69 examples on average for learning class distribution

Table 8
Overall pairwise difference in means Tukey's post hoc test for cost factor.

Cost	Subset 1	Subset 2	Subset 3
Zero	44.35		
Low		61.70	
High			87.41

Mean square error = 156.974; Harmonic mean sample size = 300.

PDFs, with a minimum of 36 examples in one training dataset and maximum of 100 examples in another. Thus, the DEA-PNN rejected over 66% of the training dataset examples from learning class PDFs. Table A1 in the Appendix A provides the breakdown of examples used by the DEA-PNN for each of its five training datasets and three data distributions. The second column shows the number of examples in Class 1 of the training dataset that received efficiency score values of 1. The third column shows the number from Class 0 that received efficiency score values of 1. The fourth column shows the other high-efficiency score value examples that were retained by the DEA-PNN, with their efficiency score values varying from between less than one and greater than or equal to the threshold value of 0.9. Lastly, the fifth column is sum of the numbers appearing in the preceding three columns and shows the total number of examples considered in learning PDFs for the DEA-PNN.

Simulated dataset results seem to indicate that the DEA-PNN may be a promising technique to use when dataset class distributions are either exponential or uniform. However, the SVM may be more appropriate when dataset class distributions are normal. As mentioned earlier, simulated datasets represent ideal distributions, representativeness assumption, equal class priors, and variable independence conditions that are rarely observed in the real world. Real-world datasets do not adhere to these ideal conditions and may contain distributions that are only approximations of ideal statistical distributions. In our simulated datasets, decision-making variables were generated independently without any correlations. Real-world decision-making variables may have positive or negative correlations between them and are very different from simulated datasets. Additionally, the number of examples belonging to both classes in real-world datasets may not be equal. With unequal examples, and sometimes smaller datasets, sampling may lead to a violation of the representativeness assumption as well.

We tested our techniques on two real-world datasets from the domain of corporate bankruptcy prediction. These datasets were generated and used in a previous study (Pendharkar & Nanda, 2006). There are two corporate bankruptcy datasets. The first dataset contains 100 examples, with 50 firms that filed for bankruptcy and 50 that did not, between the years 1987 and 1992. The second dataset contains 83 firms, where 22 of these filed for bankruptcy and the remaining 61 firms were financially solvent, from the years 1993 to 1995. Both datasets contained five decision-making variables (ratios):

- *Earnings Before Interest and Taxes/Interest Expense*,
- *Earnings Before Interest and Taxes/Assets*,
- *Current Assets/Current Liabilities*,
- *Retained Earnings/Assets*,
- *Market Value of Equity/Book Value of Debt*.

Since the five ratios had higher values for non-bankrupt firms, we gave them a class label of "1", and assigned "0" to bankrupt firms. Some ratios were negative so we processed the data using the method described in Section 2 to move data into the positive quadrant. The two datasets represent an interesting scenario in terms of class bias, where the percentage of examples belonging to both classes is balanced in the first dataset and unbalanced in the second.

Given the already small sample sizes of our real-world datasets, splitting them into two for training and test purposes would further reduce the sample size. Under such circumstances, we decided to use V-fold experimentation. In a V-fold sampling, the original dataset is divided into five split datasets of approximately equal size (five split-datasets of 20 examples each for the first dataset). We made sure that class balances are not impacted in the split datasets (for example, in the first dataset, each split dataset contained 10 instances each of bankrupt non-bankrupt firms). Once five split datasets are formed, V-fold sampling uses four of them to create a training dataset, and the fifth as a test dataset. Five unique training and test experiments can be performed using V-fold sampling, where each example appears either in a training or a test dataset in different experiments.

Table 9
Holdout sample misclassification cost for first dataset (1987–1992).

Fold	DEA-PNN	EDEA	HNDEA	PNN	SVM
Zero misclassification cost asymmetry					
1	6	14	14	0	15
2	18	7	7	20	13
3	16	4	4	20	12
4	7	8	7	11	8
5	6	6	6	0	6
Average	**10.6**	**7.8**	**7.6**	**10.2**	**10.8**
Low misclassification cost asymmetry					
1	20	16	16	20	16
2	10	10	8	0	20
3	2	6	6	0	20
4	14	13	13	14	6
5	19	7	7	20	6
Average	**13**	**10.4**	**10**	**10.8**	**13.6**
High misclassification cost asymmetry					
1	20	16	16	20	12
2	0	16	16	0	20
3	0	12	12	0	20
4	18	24	24	18	9
5	19	8	8	19	12
Average	**11.4**	**15.2**	**15.2**	**11.4**	**14.6**

Tables 9 and 10 report test dataset results for each fold, their means, and paired t-test results on the difference in means between different techniques. While the difference in means between techniques was not statistically significant, it is important to note that the mean value of DEA-PNN is higher than that of PNN for zero and low misclassification cost asymmetry. Since DEA-PNN is a hybrid technique that selects only a subset of examples selected by PNN, it appears that it had too few examples to learn class PDFs in bankruptcy prediction datasets. Table A2 in the Appendix A provides a breakdown of the number of examples selected by the DEA-PNN to learn its class PDFs. This number was 39 on average. The PNN, on the other hand, used all 80 training examples to learn its PDFs. The results appear to suggest that the selection threshold

Table 10
Pairwise |t|-value for the difference in means for first dataset (1987–1992).

	EDEA	HNDEA	PNN	SVM
Zero misclassification cost asymmetry				
DEA-PNN	0.73	0.79	0.17	0.08
EDEA		1	0.42	1.79
HNDEA			0.46	2.00
PNN				0.13
Low misclassification cost asymmetry				
DEA-PNN	0.97	1.15	1.09	0.10
EDEA		0.19	0.47	0.83
HNDEA			0.21	0.89
PNN				0.39
High misclassification cost asymmetry				
DEA-PNN	0.75	0.75	0	0.46
EDEA		0	0.75	0.14
HNDEA			0.75	0.14
PNN				0.46

Table 11
Holdout sample misclassification cost for second dataset (1993–1995).

Fold	DEA-PNN	EDEA	HNDEA	PNN	SVM
Zero misclassification cost asymmetry					
1	7	6	6	11	7
2	9	6	5	13	9
3	8	5	5	14	8
4	10	10	10	11	9
5	8	4	6	14	5
Average	**8.4**	**6.2**	**6.4**	**12.6**	**7.6**
Low misclassification cost asymmetry					
1	6	8	8	6	11
2	12	5	5	12	11
3	10	6	7	8	10
4	10	13	13	12	9
5	9	8	8	11	10
Average	**9.4**	**8**	**8.2**	**9.8**	**10.2**
High misclassification cost asymmetry					
1	6	10	10	6	12
2	16	5	5	16	12
3	12	9	10	8	12
4	10	15	15	10	13
5	11	13	13	11	14
Average	**11**	**10.4**	**10.6**	**10.2**	**12.6**

Table 12
Pairwise |t|-value for the difference in means for second dataset (1993–1995).

	EDEA	HNDEA	PNN	SVM
Zero misclassification cost asymmetry				
DEA-PNN	2.99*	2.82*	4.58*	1.37
EDEA		0.40	4.00*	1.87
HNDEA			4.23*	1.18
PNN				4.22*
Low misclassification cost asymmetry				
DEA-PNN	0.75	0.66	0.53	0.72
EDEA		1.00	1.12	1.30
HNDEA			1.00	1.21
PNN				0.28
High misclassification cost asymmetry				
DEA-PNN	0.20	0.13	1.000	0.95
EDEA		1.00	0.07	1.50
HNDEA			0.14	1.38
PNN				1.42

* Significant at 95% level of statistical significance.

of 0.9 used by the DEA-PNN may have led to under sampling. In both cases the difference in means between techniques is not significant, but it is important to note that sample size and threshold selection do play a role in the performance of DEA-PNN. Performance degradation may occur due to low sample size and/or due to a violation of the representativeness assumption.

Tables 11 and 12 report the results for the second dataset. The results indicate that the PNN had the worst statistically significant performance when misclassification cost asymmetry was zero. Furthermore, the EDEA and the HNDEA performed better than the DEA-PNN under the same zero asymmetry. However, all techniques performed statistically similarly when the misclassification cost asymmetry was low or high. Table A3 in the Appendix A provides a breakdown of the number of examples selected by the DEA-PNN to learn its class PDFs. On average, the DEA-PNN used about 51 examples. In contrast, the first fold of PNN contained 88 training examples and the other four folds contained 84 training examples.

In comparison to the simulated dataset results, the EDEA and the HNDEA results for real-world datasets were much better. One reason for this improvement may be the problem domain. It is well known in bankruptcy prediction literature, that class distributions for these problems are similar to exponential distributions (Lee, 2014), and techniques that perform well on exponential distributions generally do well for bankruptcy prediction problems (Ono, 2006) also.

5. Summary, discussion, and directions for future work

Our study indicates that probabilistic DEA techniques may hold the promise of improved classification results in certain focused classification problem domains. This DEA classification niche requires a linearly inseparable classification problem with continuous decision-making attributes. Additionally, monotonicity, where higher values of decision-making attributes lead to classification into the class label of 1, and exponential class distributions may be desired. Generally, the DEA-PNN technique performs as well as or better than competing misclassification cost-sensitive SVM. However, the DEA-PNN can select too few examples to learn its class distribution PDFs when training data sample sizes are small (fewer than 100 examples). Using traditional PNN as a benchmark allows a decision-maker to detect if training sample sizes are too small, and may, therefore, impact the performance of DEA-PNN. Thus, a decision-maker may use DEA-PNN and PNN results together and improve classification accuracy by making adjustments to the DEA-PNN threshold so that the results it produces are always better than or equal to those of the PNN.

In a classification problem where all decision-making variables are categorical, the DEA technique should not be used. However, for mixed decision-making attributes, where some variables are continuous and others are categorical, a few modifications can be made to incorporate categorical variables. First, if these categorical variables are ordinal, then Banker and Morey's (1986) non-controllable models can be used to incorporate them along with other continuous variables. If all categorical variables are non-ordinal and binary, then these binary variables can be relabeled using a methodology that is similar to the class relabeling described in Section 3. For categorical variables that are neither binary nor ordinal, training data needs to be split for each category of the variable, and separate DEA models may be built for each category. Such an analysis is combinatorial in nature and requires large size datasets so that training datasets for each category have linear inseparable classification problems. In practice, non-ordinal categorical variables, along with continuous variables, may impose a limit on the use of DEA models for classification problems. For such problems, a better approach may be to use hybrid techniques that process categorical variables and continuous variables separately. Perhaps use of decision trees for categorical variables, and DEA for continuous variables may be an option for such a hybrid technique. Future research is needed in this area.

Appendix A

See Tables A1–A3.

Table A1
Breakdown of examples used to learn DEA-PNN PDFs.

Dataset	Class 1 efficient	Class 0 efficient	Inefficient	Total
Normal distribution				
1	23	27	50	100
2	18	26	31	75
3	14	25	50	89
4	21	21	26	68
5	10	18	44	72
Exponential distribution				
1	32	12	1	45
2	38	13	1	52
3	17	17	2	36
4	26	12	2	40
5	43	17	0	60
Uniform distribution				
1	39	35	17	91
2	32	26	14	72
3	26	35	19	80
4	37	23	19	79
5	27	24	22	73

Table A2
Breakdown of examples used to learn DEA-PNN pdfs for the first dataset.

Training dataset fold	Class 1 efficient	Class 0 efficient	Inefficient	Total
1	11	5	30	46
2	7	9	15	31
3	8	9	17	34
4	8	9	14	31
5	11	6	36	53

Table A3
Breakdown of examples used to learn DEA-PNN PDFs for the second dataset.

Training dataset fold	Class 1 efficient	Class 0 efficient	Inefficient	Total
1	4	9	39	52
2	4	8	40	52
3	3	7	40	50
4	6	7	35	48
5	3	9	39	51

References

Banker, R. D. (1993). Maximum likelihood, consistency and data envelopment analysis: A statistical foundation. *Management Science, 39*(10), 1265–1273.

Banker, R. D. (1996). Hypothesis tests using data envelopment analysis. *Journal of Productivity Analysis, 7*(2), 139–159.

Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management Science, 30*, 1078–1092.

Banker, R. D., Kauffman, R. J., & Morey, R. C. (1990). Measuring gains in operational efficiency from information technology: A study of the positran deployment at Hardee's Inc. *Journal of Management Information Systems, 7*(2), 29–54.

Banker, R. D., & Morey, R. C. (1986). The use of categorical variables in data envelopment analysis. *Management Science, 32*(12), 1613–1627.

Bose, I., & Chen, X. (2009). Quantitative models for direct marketing: A review from systems perspective. *European Journal of Operational Research, 195*, 1–16.

Bult, J. R., & Wittink, D. R. (1996). Estimating and validating asymmetric heterogeneous loss functions applied to health care fund raising. *International Journal of Research in Marketing, 13*, 215–226.

Duda, R. O., & Hart, P. (1973). *Pattern classification and scene analysis.* New York, NY: Wiley.

Dyson, R. G., & Thannassoulis, E. (1988). Reducing weight flexibility in data envelopment analysis. *Journal of Operational Research Society, 39*(6), 563–576.

Emrouznejad, A., & Latef, A. A. (2010). A semi-oriented radial measure for measuring the efficiency of decision making units with negative data. *European Journal of Operational Research, 200*, 297–304.

Lee, M.-C. (2014). Business bankruptcy prediction based on survival analysis approach. *International Journal of Computer Science & Information Technology, 6*(2), 103–119.

Lovell, C. A. K., & Pastor, J. T. (1999). Radial DEA models without inputs or without outputs. *European Journal of Operational Research, 118*(1), 46–51.

Masoumzadeh, A., Toloo, M., & Amirteimoori, A. (2016). Performance assessment in production systems without explicit inputs: An application to basketball players. *IMA Journal of Management Mathematics, 27*(2), 143–156.

Ono, T. (2006). Default prediction using double-exponential distribution and empirical evidence with listed US companies. In *Stanford-Tsukuba Joint Workshop on Financial Engineering and Systems Management* (pp. 1–14). Palo Alto, CA.

Parzen, E. (1962). On estimation of probability density function and mode. *Annals of Mathematical Statistics, 33*, 1065–1076.

Pendharkar, P. C. (2002). A potential use of data envelopment analysis for the inverse classification problem. *Omega-International Journal of Management Science, 30*(3), 243–248.

Pendharkar, P. C. (2011a). A hybrid radial basis function and data envelopment analysis neural network for classification. *Computers & Operations Research, 38*(1), 256–266.

Pendharkar, P. C. (2012). Fuzzy classification using the data envelopment analysis. *Knowledge-Based Systems, 31*, 183–192.
Pendharkar, P. C. (2015). Linear models for cost-sensitive classification. *Expert Systems, 32*(5), 622–636.
Pendharkar, P. C., & Troutt, M. D. (2014). Interactive classification using data envelopment analysis. *Annals of Operations Research, 214*(1), 125–141.
Pendharkar, P. C. (2011b). Probabilistic approaches for credit screening and bankruptcy prediction. *Intelligent Systems in Accounting, Finance and Management, 18*(4), 177–193.
Pendharkar, P. C., & Troutt, M. D. (2011). DEA based dimensionality reduction for classification problems satisfying strict non-satiety assumption. *European Journal of Operational Research, 212*(1), 155–163.
Pendharkar, P., & Nanda, S. (2006). A misclassification cost-minimizing evolutionary-neural classification approach. *Naval Research Logistics, 53*(5), 432–447.
Pendharkar, P. C. (2005). A data envelopment analysis-based approach for data preprocessing. *IEEE Transactions on Knowledge and Data Engineering, 17*(10), 1379–1388.
Specht, D. (1990). Probabilistic neural networks. *Neural Networks, 3*, 109–118.
Sueyoshi, T. (1999). DEA-discriminant analysis in the view of goal programming. *European Journal of Operational Research, 115*(3), 564–582.
Sueyoshi, T. (2001). Extended DEA-discriminant analysis. *European Journal of Operational Research, 131*(2), 324–351.
Sueyoshi, T. (2004). Mixed integer programming approach of extended DEA-discriminant analysis. *European Journal of Operational Research, 152*, 45–55.
Toloo, M. (2013). The most efficient unit without explicit inputs: An extended MILP-DEA model. *Measurement, 46*(9), 3628–3634.
Toloo, M., & Kresta, A. (2014). Finding the best asset financing alternative: A DEA-WEO approach. *Measurement, 55*, 288–294.
Toloo, M., Saen, R. F., & Azadi, M. (2015). Obviating some of the theoretical barriers of data envelopment analysis-discriminant analysis: An application in predicting cluster membership of customers. *Journal of Operational Research Society, 66*(4), 674–683.
Toloo, M., & Tavana, M. (2017). A novel method for selecting a single efficient unit in data envelopment analysis without explicit inputs/outputs. *Annals of Operations Research, 253*(1), 657–681.
Toloo, M., Zandi, A., & Emrouznejad, A. (2015). Evaluation efficiency of large-scale data set with negative data: an artificial neural network approach. *The Journal of Supercomputing, 71*(7), 2397–2411.
Troutt, M. D., Rai, A., & Zhang, A. (1994). Use of Data Development Analysis for Certain Case Based Expert System Applications. In *Proceedings of international conference of information systems* (pp. 355–362).
Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques (Third)*. San Francisco, California: Morgan Kaufmann Publishers Inc.
Zhang, Y., Yang, C., Yang, A. R., Xiong, C., Zhou, X. C., & Zhang, Z. G. (2015). Feature selection for classification with class-separability strategy and data envelopment analysis. *Neurocomputing, 166*, 172–184.
Zhao, Y., Zhao, Y., & Song, I. (2009). Predicting new customers' risk type in the credit card market. *Journal of Marketing Research, XLVI*, 506–517.