Ann Oper Res
DOI 10.1007/s10479-009-0669-2

Applications and extensions of cost curves to marine container inspection

R. Hoshino · D. Coughtrey · S. Sivaraja · I. Volnyansky · S. Auer · A. Trichtchenko

Abstract Drummond and Holte introduced the theory of cost curves, a graphical technique for visualizing the performance of binary classifiers over the full range of possible class distributions and misclassification costs. In this paper, we use this concept to develop the Improvement Curve, a new performance metric for predictive models. Improvement curves are more user-friendly than cost curves and enable direct inter-classifier comparisons. We apply improvement curves to measure risk-assessment processes at Canada's marine ports. We illustrate how implementing even a basic predictive model would lead to improved efficiency for the Canada Border Services Agency, regardless of class distributions or misclassification costs.

1 Motivation

Since its inception in December 2003, the Canada Border Services Agency (CBSA) has gradually moved towards a "smart border", where day-to-day operational decisions are increasingly guided by science and technology rather than the "gut feelings" of customs officers. In this paper, we describe a specific example of an operational activity, where the existing risk management strategy relating to marine container inspection can be improved, saving CBSA both time and money.

Fumigants are widely used in the shipping industry to kill invasive alien species found inside the wooden pallets of marine cargo containers. These wooden pallets provide temporary sanctuary to insects and other alien invasive animals that can be harmful to human health, the environment, and our agriculture. Fumigants are chemical compounds in the form of gases that are typically used as insecticides to kill these alien species. While these fumigants serve an important function, the chemicals may pose a serious risk to the health and safety of CBSA officers if the marine containers are not properly ventilated. As a result,

R. Hoshino (✉) · D. Coughtrey · S. Sivaraja · I. Volnyansky · S. Auer · A. Trichtchenko
Canada Border Services Agency, Government of Canada, 79 Bentley Avenue, Ottawa, Ontario, K2E 6T7, Canada
e-mail: richard.hoshino@gmail.com

Published online: 07 November 2009

CBSA has implemented an agency-wide policy to test all marine containers for fumigation prior to examination.

As a result of this policy, every single container referred by CBSA officers for further examination is subject to a test prior to inspection, where a chemical analysis is performed to measure the levels of methyl bromide, phosphine, and other fumigants. If any of these chemicals are detected at a concentration level above a fixed threshold, then the container is ventilated to remove these hazardous fumigants, and then re-tested to ensure that the container is safe to examine. Once the container is deemed safe, CBSA officers open the container to check for contraband, banned weapons, and other items that might threaten the country's health and security.

While only a small percentage of marine containers is referred by officers for further examination, conducting a chemical test for fumigants can be expensive and time-consuming. Especially when there is reason to believe that a container has been fumigated, conducting a chemical test to confirm this is a waste of resources and could cause a significant backlog at our marine ports.

To address this issue of backlog, we use historical data to build a simple binary classifier that predicts whether a container has been fumigated, based on just four features. If the classifier predicts that a container has been fumigated, we ventilate the container without bothering to perform the initial chemical test, thus saving time and money for each correct prediction. (Conversely, for each incorrect prediction, we unnecessarily ventilate a non-fumigated container, thus wasting time and money.)

Given such a classification model, how do we measure how much money and time we would save post-implementation? That is the key question for senior decision-makers as they evaluate the merits of implementing a risk-management strategy based on data mining. A partial answer to this question can be determined from receiver operating characteristic (ROC) curves and cost curves, two common graphical techniques to evaluate performance. We will explain why neither technique is adequate, and develop a single graph that provides all of the information needed to measure the percentage improvement of replacing one strategy with another. We call this the *improvement curve*.

This paper proceeds as follows: in Sect. 2 we provide the necessary definitions for the paper, and present the cost function that is to be minimized. In Sect. 3, we explain how a classification model's performance can be determined from their ROC curves and cost curves, and discuss the limitations of each evaluation metric. In Sect. 4, we introduce improvement curves and explain its significance. We also describe how to construct this curve given a classification model. In Sect. 5, we apply the theories developed in the previous section to a real-life problem in operations research, namely the risk-assessment of marine cargo containers. We prove that existing operating procedures can be made more efficient by implementing a simple binary classifier based on just four features, and use the improvement curve to estimate how much time and money we would save post-implementation. Finally, in Sect. 6, we conclude with directions for further research.

2 Definitions

In a two-class prediction problem (i.e., a binary classification), instances are either positive (P) or negative (N). The goal is to create a classifier that correctly predicts the class of an instance to optimize decision-making. Developing reliable classifiers has been the subject of much research in the machine learning community over the past few decades, and has natural applications to operations research.

Table 1 The standard confusion matrix

	Predicted P	Predicted N
Actual P	*TP*	*FN*
Actual N	*FP*	*TN*

The performance of any classifier can be measured using a *confusion matrix*, where the rows represent the actual class of an instance and the columns represent the predicted class.

From the confusion matrix of a binary classifier, there are four possible outcomes: true positives (*TP*), false positives (*FP*), true negatives (*TN*), and false negatives (*FN*). In practice, the numbers in the confusion matrix are obtained by applying a given classifier to a test set of independent instances and counting how many fall into each of the four categories. In the literature, false positives are known as Type I errors and false negatives are known as Type II errors. A perfect classifier will have no Type I or Type II errors (i.e., $FP = FN = 0$).

When we evaluate a binary classifier, we look at the four outcomes from the confusion matrix and create some type of metric that measures the classifier's overall performance. One of the more common performance metrics is accuracy, which is a simple function of the four outcomes in the confusion matrix:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}.$$

The accuracy metric gives us the proportion of examples that are correctly classified by a given algorithm. It is clearly an undesirable quality for a classifier to have low accuracy. However, high accuracy is not a sufficient condition for a classifier to be effective. For example, consider a data set in which 99% percent of the data belongs to one class and 1% belongs to the other. By predicting everything as being a member of the larger class, we achieve 99% accuracy, but such a classifier is worthless: the accuracy metric completely ignores the costs of misclassification, which is a significant problem if there is a high false negative cost and a low false positive cost.

Thus, we will need something more significant to measure overall performance than this simple scalar measure that does not account for misclassification costs or class imbalances. This motivates two other evaluation metrics which are more meaningful: the ROC curve and the cost curve. Later in this paper, we introduce another graphical performance metric which we will call the improvement curve.

The true positive rate (*TPR*) is the proportion of positive containers that are correctly predicted to be positive. In other words,

$$TPR = \frac{TP}{TP + FN}.$$

The false positive rate (*FPR*) is the proportion of negative containers that are incorrectly predicted to be positive. In other words,

$$FPR = \frac{FP}{FP + TN}.$$

Similarly, we define the false negative rate (*FNR*) and the true negative rate (*TNR*). By definition, $TPR + FNR = 1$ and $FPR + TNR = 1$. Finally, we assign a cost of C^- to every false negative and a cost of C^+ to every false positive. For the purposes of this paper, we will assume that C^- and C^+ are constants (e.g., any false negative has the same misclassification cost as any other false negative). Furthermore, we will assume that $C^- > 0$ and $C^+ > 0$. From an operations research perspective, the goal is to develop a classifier that minimizes the total misclassification cost M, given by the following function:

$$M = FN \cdot C^- + FP \cdot C^+.$$

In our application of risk-managing fumigated containers (Sect. 5), a fumigated container will be considered "positive". Thus, a false positive occurs when the classifier incorrectly predicts that a non-fumigated container has been fumigated, leading to an unnecessary ventilation. And a false negative occurs when the classifier fails to predict that a fumigated container has been fumigated, leading to an unnecessary chemical test. Thus, C^+ denotes the cost of a ventilation and C^- the cost of a chemical test.

3 ROC and cost curves

In this paper, we examine a subset of predictive models known as *classification models*. Classification models partition a data set into distinct "classes", and assign some "score" to each class. For example, binary trees are classification models, as each instance falls into a unique leaf node (representing a class such as "A is true and B is true and C is false"). Furthermore, every instance falling into this leaf node is assigned the same score. Another example of a classification model is one derived from n binary variables and 2^n classes, where each class is a unique n-tuple of 0s and 1s. If two instances have the same n-tuple, they fall into the same class, and hence, they will be given the same score.

In this section, we discuss ROC curves and cost curves for classification models. We explain how to construct each curve from a given model, and describe how to turn the model into the binary classifier that minimizes the cost function $M = FN \cdot C^- + FP \cdot C^+$. We will discuss the limitations of each approach in the context of real-world decision-making as a motivation for the theory of improvement curves.

3.1 ROC curves

A receiver operating characteristic (ROC) curve has two dimensions, with the *FPR* on the x-axis and the *TPR* on the y-axis. ROC curves are generated from classification models that rank classes in order of risk, often via some scoring algorithm. These models turn into binary classifiers by setting a particular threshold: any class appearing above the threshold is predicted positive; otherwise it is predicted negative. For each possible threshold, one calculates the algorithm's *TPR* and *FPR* from the corresponding confusion matrix; this information is then collected to draw the ROC curve.

While ROC analysis has been around for 60 years, it was only introduced to the machine learning community in the last decade (Provost and Fawcett 2001). The ROC curve became a popular method to evaluate the effectiveness of a predictive model, as it provides a simple graphical representation of overall performance. At each point on the curve, one can compare the trade-off between false positives and true positives. Lowering the discrimination threshold will yield more true positives, but at the cost of generating more false positives. A detailed discussion of ROC curves and its applications to data mining has previously appeared in the literature (c.f., Flach 2003, 2004).

Table 2 provides a model with 8 classes, which assigns a level of risk (ranked from highest to lowest) to $2^3 = 8$ tuples representing all possible combinations of responses to

Table 2 A 3-variable classification model

i	Q_1	Q_2	Q_3	p_i	n_i	Risk
1	Y	Y	Y	10	10	Highest
2	Y	N	Y	15	30	⇓
3	Y	N	N	15	40	⇓
4	N	Y	Y	15	50	⇓
5	Y	Y	N	20	100	⇓
6	N	Y	N	10	60	⇓
7	N	N	Y	10	70	⇓
8	N	N	N	5	140	Lowest

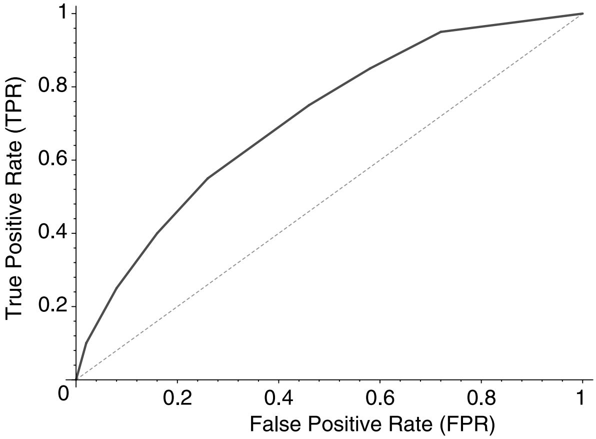


Fig. 1 ROC curve for the classification model

three Yes-No questions. For each class (labeled by the indices $i = 1, 2, \ldots, 8$), the number of positive and negative instances (p_i and n_i) is tabulated. Figure 1 gives the ROC curve for this classification model.

For example, suppose that an instance is predicted positive if and only if its class appears in the highest two risk categories. Under this classifier, the confusion matrix is $TP = 25$, $FP = 40$, $FN = 75$ and $TN = 460$. It follows that $FPR = \frac{40}{40+460} = 0.08$ and $TPR = \frac{25}{25+75} = 0.25$. This classifier corresponds to the point $(FPR, TPR) = (0.08, 0.25)$ on the ROC curve.

Repeating this process for every possible threshold, we obtain a total of nine points, representing the (FPR, TPR) coordinates. These nine points are then joined to create the ROC curve. Note that every ROC curve will include the points $(0, 0)$ and $(1, 1)$, corresponding respectively to the trivial cases where the classifier predicts everything to be negative and everything to be positive. The baseline scenario is a straight line from $(0, 0)$ to $(1, 1)$ which represents a risk-scoring algorithm with zero discrimination capacity. In particular, the expected ROC curve of an algorithm that assigns random risk scores to each instance would be this straight diagonal line.

One of the most common ways to measure a predictive model's performance is to calculate the *area under the ROC curve*, denoted by ROC AUC. The ROC AUC is widely used throughout the machine learning community to evaluate and compare risk-scoring effectiveness (Flach 2003). The AUC is preferred over the Accuracy function defined in Sect. 1, as the AUC provides a measure of discrimination accuracy that is independent of any specific threshold. This is because the AUC corresponds to the shape of the entire ROC curve rather than any single ROC point (Swets et al. 2000).

A perfect ROC curve has an AUC of 1, which occurs when the curve includes the point $(FPR, TPR) = (0, 1)$. In Fig. 1, the ROC AUC is 0.71. According to one criteria (Swets 1988), a poor model will have a ROC AUC between 0.5 and 0.7, a moderate model between 0.7 and 0.9, and an excellent model above 0.9. Note that a ROC AUC of 0.5 is no better than an algorithm that assigns scores at random.

It turns out that the ROC AUC is equivalent to the probability that the algorithm will rank a randomly chosen positive instance higher than a randomly chosen negative instance (Bradley 1997). This is equivalent to the Wilcoxon test of ranks, also known as the Mann-Whitney U statistic (Hanley and McNeil 1982).

The ROC curve in Fig. 1 is convex. It is straightforward to show that convexity is attained by ranking the classes in decreasing order of the ratio $\frac{p_i}{n_i}$. This well-known result has been cited in previous papers (c.f., Flach 2004; Boström 2005, 2007). The ROC curve of any scoring algorithm based on a classification model can be made convex by arranging the resulting classes in this order. A ROC curve clearly attains its maximum AUC when it is convex.

Note that convexity holds just for the *training set* used to create the model. Naturally, when evaluating the given model on an independent *testing set*, its ROC curve is not necessarily convex.

Suppose we have two different classification models, each producing a convex ROC curve after arranging the classes in the correct order. How do we determine which of these models is superior? If one ROC curve dominates the other (i.e., it is above and to the left of the other ROC curve), then the question is trivial. However, if the two ROC curves cross, then one model is superior in some circumstances and inferior in others.

When two ROC curves cross, a single scalar measure such as the ROC AUC may be insufficient in determining which model should be adopted. Especially if the cost of false positives is significantly different from the cost of false negatives, then a predictive model with a lower AUC may produce a binary classifier that has a lower total misclassification cost than a model with a higher AUC.

We now explain how the optimal misclassification cost of a binary classifier can be determined directly from the ROC curve. The following proposition explains how to find the point (FPR, TPR) on the ROC curve that corresponds to the optimal binary classifier that minimizes the misclassification cost.

Proposition 3.1 (Provost and Fawcett 1997) *Let P and N be the number of positive and negative instances, respectively. By definition, $P = TP + FN$ and $N = FP + TN$.*

Consider the family of lines with slope $t = \frac{P \cdot C^-}{N \cdot C^+}$. Let k be the largest number for which the line $y = \frac{x}{t} + k$ intersects a point (x', y') on the ROC curve. Then, the binary classifier that minimizes the total misclassification cost occurs when $FPR = x'$ and $TPR = y'$.

For this number k, it is straightforward to show that the misclassification cost is $M = (1 - k) \cdot P \cdot C^-$. As long as we can estimate t, we can determine the optimal classifier of a model and calculate its misclassification cost.

The family of lines with slope $\frac{1}{t}$ is an example of an *isometry*. Many optimization problems involving ROC curves can be solved by considering isometries (Flach 2003; Furnkranz and Flach 2003), which are also known as *iso-performance lines* (Drummond and Holte 2006).

While Proposition 3.1 enables us to determine the optimal point on the ROC curve to create the binary classifier that minimizes the total misclassification cost, the process is quite cumbersome. Furthermore, when we compare two different predictive models, it is not obvious which model is better if their ROC curves cross. In order to resolve the question, one must construct iso-performance lines to determine each model's optimal classifier, and then compare the resulting misclassification costs to determine which model is better.

3.2 Cost curves

Motivated by the limitations of ROC curves, let us explore *cost curves* (Drummond and Holte 2000a, 2000b, 2004, 2006), which do a much better job of representing the performance of a scoring algorithm. The authors compare ROC curves to cost curves, and show that with the latter, each of the following questions can be answered instantly by visual inspection, while simultaneously retaining nearly all of the attractive features of ROC curves (Drummond and Holte 2006). They also note that none of these four questions can be answered for ROC curves by visual inspection:

(a) What is a classifier's performance (expected cost) given specific misclassification costs and class probabilities?
(b) For what misclassification costs and class probabilities does a classifier outperform the trivial classifiers that assign all examples to the same class?
(c) For what misclassification costs and class probabilities does one classifier outperform another?
(d) What is the difference in performance between two classifiers?

Cost curves are powerful as they measure expected cost across all possible class distributions, and take misclassification costs into account.

The cost curve of a predictive model is a two-dimensional plot of the "probability times cost" value against its "normalized expected cost". We now briefly define both of these terms.

The x-axis of a cost curve is a function of the number of positive and negative instances, combined with the costs of a false positive and a false negative. In this light, the x-axis is a function of four unknown variables: P, N, C^+, and C^-. We collapse these four unknowns into a single variable $PC(+)$, denoted as *probability times cost*.

First we define $t = \frac{P \cdot C^-}{N \cdot C^+}$, which is the same function that was introduced in Proposition 3.1. Now define

$$PC(+) := \frac{P \cdot C^-}{P \cdot C^- + N \cdot C^+} = \frac{t}{t+1}.$$

This is the x-axis of the cost curve, which ranges from 0 to 1, as t can take on any positive value.

The y-axis is the *normalized expected cost*, which is the normalized value of the total misclassification cost so that the output is between 0 and 1. From Sect. 1, the misclassification cost is given by the formula

$$\begin{aligned} M &= FN \cdot C^- + FP \cdot C^+ \\ &= FNR \cdot P \cdot C^- + FPR \cdot N \cdot C^+, \end{aligned}$$

where P and N are the number of positive and negative instances, respectively. The misclassification cost is maximized when the classifier predicts every instance incorrectly (i.e.,

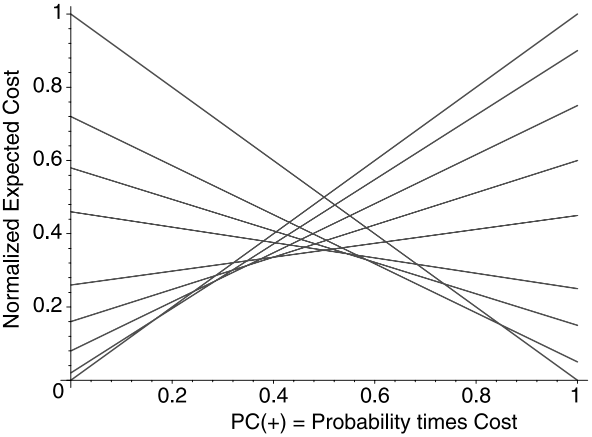


Fig. 2 Cost lines for all 9 ROC points

$FPR = FNR = 1$). In this extreme case, the misclassification cost is $P \cdot C^- + N \cdot C^+$. Dividing M by this value, the normalized expected cost is defined to be

$$N(E[cost]) := \frac{FNR \cdot P \cdot C^- + FPR \cdot N \cdot C^+}{P \cdot C^- + N \cdot C^+}.$$

This is the y-axis of the cost curve. By this definition, the perfect classifier (i.e., $FPR = FNR = 0$) has $N(E[cost]) = 0$ and the worst possible classifier has $N(E[cost]) = 1$.

In order to construct a cost curve, we require the following proposition which explains the connection between cost space and ROC space.

Proposition 3.2 (Drummond and Holte 2006) *There is a bi-directional point-line duality between ROC space and cost space. In other words, a point in ROC space is represented by a line in cost space, and a line in ROC space is represented by a point in cost space, and vice-versa. Specifically, the point* (p, q) *in ROC space corresponds to the line* $y = (1 - q - p)x + p$ *in cost space.*

By this proposition, each point on a ROC curve corresponds to a cost line. The *cost curve* is then formed by constructing the *lower envelope* of these cost lines. To illustrate the construction of a cost curve, consider the classification model given in Table 2 whose ROC curve was presented in Fig. 1.

In Fig. 1, $(FPR, TPR) = (0.08, 0.25)$ is a point on the ROC curve. By Proposition 3.2, this point on the ROC curve corresponds to the cost line joining $(0, 0.08)$ to $(1, 0.75)$.

We now repeat the process for every other point on the ROC curve, each corresponding to a binary classifier that can be generated from this classification model. Since there are nine points on the ROC curve, there are nine cost lines in total. All nine cost lines are drawn in Fig. 2.

Having generated the cost lines for the classification model, we can now construct its cost curve. The cost curve is defined to be the lower envelope of these nine lines, which is formed by taking the smallest y-value on any of the given cost lines for each possible value of x. The cost curve of this classification model is given in Fig. 3. When there are many cost lines, the lower envelope is easy to visualize.

The diagonal lines above the cost curve in Fig. 3 represent the trivial classifiers where everything is predicted negative (the cost line $y = x$), and everything is predicted positive

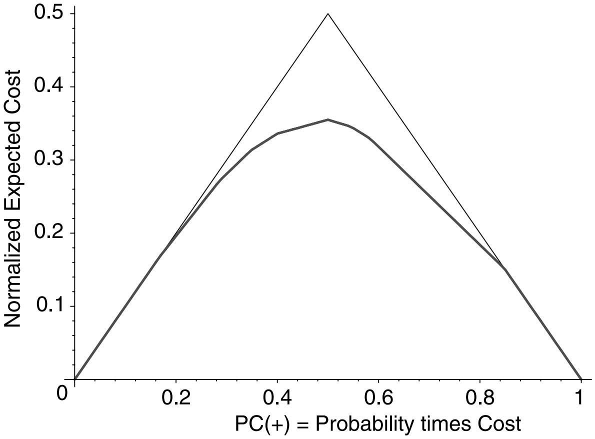


Fig. 3 The cost curve for the classification model

(the cost line $y = 1 - x$). By the definition of a lower envelope, the cost curve must be on or below these two intersecting diagonal lines.

Now that we have a cost curve, we would like to know how to apply it. As we did in the previous section with ROC curves, we estimate the value of $t = \frac{P \cdot C^-}{N \cdot C^+}$. The value of $x = PC(+)$ is then easy to calculate, since $x = \frac{t}{t+1}$. Having determined x, we determine the y-coordinate on the cost curve for this value of x.

By Proposition 3.2, the cost line that passes through this point (x, y) translates to a point (FPR, TPR) in ROC space. From this ROC point, we can determine the desired optimal binary classifier. Furthermore, the cost curve tells us exactly what the misclassification cost is: since the y-coordinate is just the normalized expected cost for this optimal classifier, the minimum total misclassification cost is simply

$$M = N(E[cost]) \cdot (P \cdot C^- + N \cdot C^+).$$

To evaluate the performance of a predictive model, it is much easier to use cost curves than ROC curves, since there is no need to construct iso-performance lines. Unlike ROC curves, the y-coordinate of the cost curve tells us the normalized expected cost, and so we can determine the optimal misclassification cost without having to calculate the y-intercept of an iso-performance line.

Furthermore, iso-performance lines are not necessary to determine which of two different predictive models is superior. As with ROC curves, if one cost curve dominates the other (i.e., one cost curve is always on or below the other), then the problem is trivial. But if the cost curves cross, the analysis is just as easy since the y-axis is just the normalized expected cost. To illustrate, assume that cost curve C_1 dominates C_2 for each $0 \leq x < c$ and cost curve C_2 dominates C_1 for each $c < x \leq 1$. If $PC(+)$ is less than c, then the optimal binary classifier is found from the first model; otherwise the optimal binary classifier is found from the second model.

Unpredictable and changing factors affect a classifier's usefulness in practice. These factors include the costs of false positives and false negatives, as well as the distribution of data to which the classifier will be applied. In many practical settings, these factors cannot be determined at the time classifiers are being evaluated and compared, and these factors often change with time (Holte 2006). That is why cost curves are so useful, as they measure performance over all possible values of $PC(+)$.

As cost curves combine four unknown variables into one, we do not require the exact values of P, N, C^{+}, or C^{-} to measure the performance of a predictive model. However, to turn the model into a binary classifier, one would require an approximate value for $x = PC(+)$ to determine the optimal classifier that minimizes $y = N(E[cost])$. But by the definition of $PC(+)$, we only need an estimate for the ratios $\frac{P}{N}$ and $\frac{C^{-}}{C^{+}}$.

Of course, by combining four variables into one, the cost curve will not allow us to measure the exact misclassification cost (recall that we had to multiply our normalized y-value by $P \cdot C^{-} + N \cdot C^{+}$). However, when comparing two predictive models, it is not necessary to have absolute costs; the normalized expected cost suffices.

Despite the strength of cost curves, there are still two limitations:

(a) Given the value of $x = PC(+)$, it is a cumbersome process to determine the optimal binary classifier of the model: first we need to calculate the y-coordinate on the cost curve, and then determine the cost line passing through this point (x, y). We then need to compute the ROC curve point (FPR, TPR) that is the dual of this cost line, and use this to find the desired threshold so that any class above the threshold is predicted positive, and any class below the threshold is predicted negative. This process is easy, but lengthy. Is there a simpler way to determine the optimal binary classifier directly from a predictive model?

(b) How does a non-mathematician interpret the cost curve? Many decision-makers would not understand the significance of "a binary classifier with a normalized expected cost of 0.05". Can the information from the cost curve be presented in a more meaningful way so that a decision-maker can readily understand the information and approve the use of data mining?

In the following section, we answer "yes" to both questions by introducing the *improvement curve*, a performance metric inspired by the cost curve but with a different y-axis. In the construction of a classification model's improvement curve, we employ normalized confidence scores. This simple scoring function will enable us to determine the optimal binary classifier directly from a predictive model, without having to generate cost lines or ROC points. This answers the first question.

The improvement curve is a user-friendly metric to measure the percentage improvement of replacing one model by another, a curve whose coordinates can be determined exactly. Thus, a decision-maker can look at the improvement curve and immediately say, "under our current operating conditions, replacing our status quo approach with a predictive model would cut by 20% the amount of money wasted by improper classification." This answers the second question.

After developing the theory of improvement curves in Sect. 4, we apply them in Sect. 5 to a real-world problem in operations research, namely the risk-management of fumigated containers at CBSA.

4 The improvement curve

Consider two classification models C_1 and C_2. For any value of $x = PC(+)$, we can determine each model's optimal binary classifier by constructing the cost curves and reading off the corresponding y-coordinates. These y-coordinates represent the lowest normalized expected costs of the two models. By definition, the superior model is the one with the lower y-coordinate. But how do we quantify the amount by which one model is superior to another?

4.1 Definition and motivation

For any value of $x = PC(+)$, let $M_1(x)$ and $M_2(x)$ be the misclassification costs of the optimal binary classifiers of C_1 and C_2, respectively. Suppose we replace classifier C_1 with classifier C_2, and we want to measure the improvement of this change. For each value of x, a natural definition for model improvement is the percentage reduction in misclassification cost:

$$I_{[C_1,C_2]}(x) := \frac{M_1(x) - M_2(x)}{M_1(x)}.$$

The improvement function $I_{[C_1,C_2]}(x)$ compares the costs of misclassification, to determine the benefit of replacing one classifier by another. This function has two obvious properties: if $C_1 = C_2$, then $M_1(x) = M_2(x)$, and so $I_{[C_1,C_2]}(x) = 0$ for all x. If C_2 is the perfect classifier with $FN = FP = 0$, then $M_2(x) = 0$ and so $I_{[C_1,C_2]}(x) = 1$ for all x.

The *improvement curve* is defined to be the two-dimensional graph plotting $x = PC(+)$ against $y = I_{[C_1,C_2]}(x) = I_{[C_1,C_2]}(PC(+))$.

Note that the x-axis of the improvement curve is the same as the x-axis of the cost curve; only the y-axes are different. The following result demonstrates a simple correlation between the y-axes of the two graphs, by showing that the improvement function can be derived directly from the cost curve function:

Proposition 4.1 *Let C_1 and C_2 be two classification models. For each $x = PC(+)$ in* [0, 1], *let $f(x)$ be the y-coordinate of the cost curve of C_1, and $g(x)$ be the y-coordinate of the cost curve of C_2.*

Then $I_{[C_1,C_2]}(x) = 1 - \frac{g(x)}{f(x)}$.

Proof Fix $0 \le x \le 1$. By definition, the minimum possible value for the misclassification cost of C_1 is $M_1(x) = f(x)(P \cdot C^- + N \cdot C^+)$ and the minimum possible value for the misclassification cost of C_2 is $M_2(x) = g(x)(P \cdot C^- + N \cdot C^+)$.

Hence, the improvement function is

$$I_{[C_1,C_2]}(x) = \frac{M_1(x) - M_2(x)}{M_1(x)} = 1 - \frac{g(x)}{f(x)},$$

which completes the proof. □

The improvement function $I_{[C_1,C_2]}(x)$ compares misclassification costs, and is superior to comparing absolute costs, as the latter lacks a point of reference. For example, suppose that the value of x is known, and the best classifier of model C_1 has an operating cost of 1.8 million dollars a year while the best classifier of model C_2 has an operating cost of 1.4 million dollars a year. If the perfect classifier costs 1.4 million dollars a year, then C_2 is perfect; if the perfect classifier costs only 0.2 million dollars a year, then the new model is far from optimal.

By introducing the improvement function as a function of misclassification costs, this issue of baseline reference is addressed: in the former case, the improvement is 100% as $M_2(x) = 0$; in the latter, it is only $\frac{1.6-1.2}{1.6} = 25\%$ as $M_1(x) = 1.8 - 0.2 = 1.6$ and $M_2(x) = 1.4 - 0.2 = 1.2$. There is a clear difference in performance improvement between these two scenarios, and we need a way to capture that. The improvement function $I_{[C_1,C_2]}(x)$ does this, as it compares misclassification costs rather than absolute costs.

Before giving an illustration of an improvement curve, we present a simple corollary:

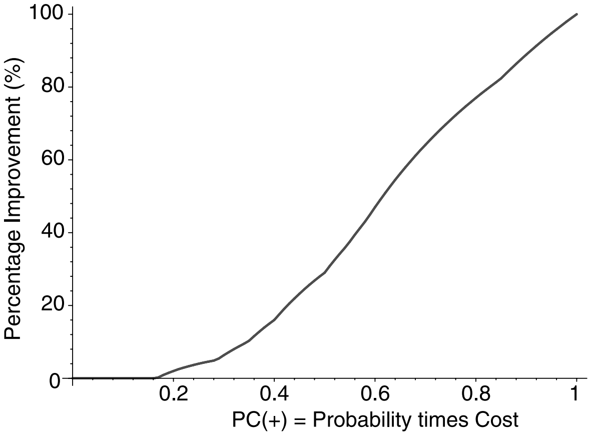


Fig. 4 Improvement curve over trivial model

Corollary 4.2 *Let C_1 and C_2 be classification models, where C_1 is the trivial model that predicts every instance to be negative. For each $x = PC(+)$ in $[0, 1]$, let $g(x)$ be the y-coordinate of the cost curve of C_2.*

Then $I_{[C_1,C_2]}(x) = 1 - \frac{g(x)}{x}$.

Proof Model C_1 is the trivial classifier that predicts every instance to be negative. Thus, $FPR = 0$ and $FNR = 1$, regardless of any score threshold. Hence, there is only one cost line, namely the line from $(0, 0)$ to $(1, 1)$. Hence, the cost curve of C_1 is simply the line $y = x$. Thus, $f(x) = x$ and the result follows from Proposition 4.1. □

To illustrate Corollary 4.2, let C_2 be the classification model in Table 2. Then, the improvement curve can be derived from the two cost curves: $f(x) = x$ for model C_1 and Fig. 3 for model C_2. The improvement curve is given in Fig. 4.

As one can see, there is a significant improvement if $x = PC(+)$ is high. Unlike cost curves, the improvement curve enables direct inter-classifier comparisons, measuring the percentage improvement of replacing one model by another. Furthermore, the improvement curve is readily accessible to a non-specialist: instead of assessing performance by comparing normalized expected costs, one can simply measure the expected performance improvement directly from the graph.

To give a concrete example, let's compare models C_1 and C_2, and suppose that $x = 0.4$. By performing the comparison with cost curves, we would have $N(E[cost]) = 0.4$ for C_1 and $N(E[cost]) = 0.336$ for C_2, leading to a reduction of 0.064 in normalized expected cost. On the other hand, a comparison with improvement curves is so much more intuitive: by Corollary 4.2, $I_{[C_1,C_2]}(x) = 1 - \frac{0.336}{0.4} = 0.16$, and so replacing model C_1 by C_2 would reduce misclassification costs by 16%.

We have now motivated the improvement curve, and why this metric would be more meaningful for decision-makers as they discern whether one strategy (or model) should be replaced by another. To complete this section, we resolve the other dilemma presented at the end of Sect. 3.2, namely the cumbersome process of determining the optimal binary classifier of a model from its cost curve. As we shall see, this process becomes trivial once we introduce a special normalized confidence function.

In addition, we will provide an explicit formula for the function $I_{[C_1,C_2]}(x)$ where C_2 is a classification model and C_1 is the trivial model that predicts every instance to be negative. In this scenario, the improvement curve can be drawn directly from a predictive model, without having to construct cost curves. We will then apply this formula in Sect. 5.

4.2 Normalized confidence function

In a classification model, the data set is partitioned into k disjoint classes, and every instance belongs to exactly one of these classes. Labeling the classes $i = 1, 2, \dots, k$, let p_i and n_i be the number of positive and negative instances appearing in class i.

Regardless of how these k classes are determined, we can rank the classes in decreasing order of the ratio $\frac{p_i}{n_i}$ to produce a convex ROC curve, as explained in Sect. 3.1. We now describe a simple risk-scoring algorithm, referred to as a *normalized confidence* function, that ranks the k classes in this desired order. To turn this risk-scoring algorithm into a binary classifier, it suffices to select a threshold; an instance is predicted positive if and only if it belongs to a class with a score above that threshold.

Instance scores (Fawcett 2001) indicate the likelihood that a given instance is positive or negative. To do this, one assigns probability scores to each class, as a function of p_i and n_i. The *confidence* c_i of class i is just the proportion of positive instances within that class:

$$c_i = \frac{p_i}{p_i + n_i}, \qquad 1 \le i \le k.$$

We generalize this idea to define a function based on the proportion of positive and negative instances, once both quantities are normalized. For each class i, define the *safety score* s_i as the following normalized confidence function:

$$s_i = \frac{\frac{n_i}{N}}{\frac{p_i}{P} + \frac{n_i}{N}}, \qquad 1 \le i \le k.$$

In other words, once the number of positive and negative instances is normalized, s_i computes the probability that an instance within that class will be negative. By definition, if $s_i = 0$, then all instances are positive; if $s_i = 1$, then all instances are negative. The higher the score, the "less risky" that class is.

Consider the model with eight classes, given in Table 2. For this model, we have

$$s_1 = \frac{\frac{10}{500}}{\frac{10}{100} + \frac{10}{500}} = \frac{1}{6}, \qquad s_8 = \frac{\frac{140}{500}}{\frac{5}{100} + \frac{140}{500}} = \frac{28}{33}.$$

The safety score function is an example of a risk-scoring algorithm, applied to the output of a classification model. In the following theorem, we prove two results. First, we show that this particular scoring algorithm guarantees a convex ROC curve, when the classes are sorted in increasing order of s_i.

Secondly, we prove that for any given point $x = PC(+)$, the optimal binary classifier can be found without analyzing cost curves or constructing iso-performance lines on ROC curves. It suffices to look at the s_i scores: if a class has an s_i score above $PC(+)$, then it is predicted negative; otherwise it is predicted positive. We will prove that this is the binary classifier that minimizes the total misclassification cost for a given $x = PC(+)$.

Theorem 4.3 *Consider a predictive model where each instance belongs to a unique class i, with $1 \le i \le k$. Define s_i to be the safety score of class i, for each $1 \le i \le k$, and sort the k classes in increasing order of s_i. Then, we have the following results.*

(a) *The ROC curve of this predictive model is convex.*
(b) *Suppose that the value of $x = PC(+)$ is known. Consider the binary classifier that predicts a class positive if and only if $s_i \leq PC(+)$. Then this binary classifier minimizes the total misclassification cost.*

Proof First, we establish the convexity of the ROC curve. In other words, we show that the safety score function s_i maximizes the ROC AUC, when the classes are sorted in increasing order by s_i score.

Recall in Sect. 3.1, we quoted the well-known result that the ROC curve is convex provided the classes are ranked in decreasing order of the ratio $\frac{p_i}{n_i}$. We show that this is equivalent to having the k classes ranked in increasing order of s_i. Letting $i < j$, we have the following chain of identical inequalities:

$$\begin{aligned}
s_i &\leq s_j, \\
\frac{\frac{n_i}{N}}{\frac{p_i}{P} + \frac{n_i}{N}} &\leq \frac{\frac{n_j}{N}}{\frac{p_j}{P} + \frac{n_j}{N}}, \\
\frac{n_i}{\frac{p_i \cdot N}{P} + n_i} &\leq \frac{n_j}{\frac{p_j \cdot N}{P} + n_j}, \\
n_i \left(\frac{p_j \cdot N}{P} + n_j \right) &\leq n_j \left(\frac{p_i \cdot N}{P} + n_i \right), \\
n_i p_j \cdot \frac{N}{P} + n_i n_j &\leq n_j p_i \cdot \frac{N}{P} + n_j n_i, \\
n_i p_j &\leq n_j p_i, \\
\frac{p_j}{n_j} &\leq \frac{p_i}{n_i}.
\end{aligned}$$

Therefore, if the classes are arranged in increasing order of s_i, then the scores are also arranged in decreasing order of the ratio $\frac{p_i}{n_i}$. This proves that the resulting ROC curve is convex, completing the proof of part (a).

Now we establish part (b). To minimize the total misclassification cost, each class must be predicted optimally. Let us compare the two possibilities:

(1) If every instance in class i is predicted positive, then $FP = n_i$ and $FN = 0$, adding $M_i = n_i \cdot C^+$ to the misclassification cost.
(2) If every instance in class i is predicted negative, then $FP = 0$ and $FN = p_i$, adding $M_i = p_i \cdot C^-$ to the misclassification cost.

Since the goal is to minimize the misclassification cost, a class should be predicted positive if and only if $p_i \cdot C^- \geq n_i \cdot C^+$. We now prove that this inequality is equivalent to $s_i \leq PC(+)$. We have the following chain of equivalent inequalities:

$$\begin{aligned}
p_i \cdot C^- &\geq n_i \cdot C^+, \\
\frac{p_i}{P} \cdot (P \cdot C^-) &\geq \frac{n_i}{N} \cdot (N \cdot C^+), \\
\frac{\frac{p_i}{P}}{\frac{n_i}{N}} &\geq \frac{N \cdot C^+}{P \cdot C^-},
\end{aligned}$$

$$\frac{\frac{p_i}{P}}{\frac{n_i}{N}} + 1 \geq \frac{N \cdot C^+}{P \cdot C^-} + 1,$$

$$\frac{\frac{n_i}{N}}{\frac{p_i}{P} + \frac{n_i}{N}} \leq \frac{P \cdot C^-}{P \cdot C^- + N \cdot C^+},$$

$$s_i \leq PC(+).$$

Therefore, we have shown that in an optimal binary classifier, a class is predicted positive if and only if $s_i \leq PC(+)$. This answers part (b). □

Having now defined safety scores, we now show how trivial it is to determine the optimal classifier of a model. Let us illustrate how to do this for the classification model given in Table 2, when $x = PC(+) = 0.4$. As a comparison, we also describe how to determine the model using cost curves; we shall see that this approach is far more cumbersome.

First, we determine the s_i scores for the model, which are given in Table 3. The s_i scores have been rounded to two decimal places.

Since $x = 0.4$, the optimal classifier of this model is found by setting the threshold at $x = 0.4$. We predict a class positive if $s_i \leq PC(+)$, and negative if $s_i > PC(+)$. Thus, the optimal classifier is the model that predicts the first four classes to be positive and the last four classes to be negative.

Now let us contrast this to how we would find the optimal classifier using cost curves. For $x = 0.4$, we find that $y = 0.336$ is the corresponding y-coordinate on the cost curve. By inspection, the cost line with endpoints $(0, 0.26)$ and $(1, 0.45)$ passes through this point. This cost line corresponds to the ROC curve point with coordinates $(FPR, TPR) = (0.26, 0.55)$. To determine the optimal classifier, we check each threshold until we find one that gives us the correct values of FPR and TPR. By inspection, having the first four classes predicted positive gives us $TPR = 0.55$ and $FPR = 0.26$, and so this is the optimal classifier.

As one can see, the cost curve technique is computationally easy, but is extremely cumbersome. By introducing safety scores, the problem becomes so much easier. Note that as an added bonus, the safety score function tells us exactly when a predictive model is better than the trivial classifiers. To illustrate, consider Fig. 3. By eyeballing the cost curve, it appears that we should predict everything negative if $PC(+) < 0.17$ and predict everything positive if $PC(+) > 0.85$. From Theorem 4.3, we can specify the exact thresholds for when these trivial classifiers are optimal: when $PC(+) < s_1 = \frac{1}{6} \sim 0.167$ and $PC(+) > s_8 = \frac{28}{33} \sim 0.849$.

Table 3 Safety scores for classification model

i	Q_1	Q_2	Q_3	p_i	n_i	s_i
1	Y	Y	Y	10	10	0.17
2	Y	N	Y	15	30	0.29
3	Y	N	N	15	40	0.35
4	N	Y	Y	15	50	0.40
5	Y	Y	N	20	100	0.50
6	N	Y	N	10	60	0.55
7	N	N	Y	10	70	0.58
8	N	N	N	5	140	0.85

4.3 Improvement function formula

To conclude this section, we provide an explicit formula for the function $I_{[C_1,C_2]}(x)$ where C_2 is a classification model and C_1 is the trivial model that predicts every instance to be negative.

To determine a formula for $I_{[C_1,C_2]}(x)$, we first require the following lemma which gives us a precise formula for the points on a model's cost curve. In this lemma, we prove that the cost curve can be determined directly from the s_i scores assigned to the k classes of the predictive model. In other words, it is not necessary to construct cost lines and form the lower envelope to generate the cost curve.

Lemma 4.4 *Consider a predictive model where each instance belongs to a unique class i, with $1 \le i \le k$. Define s_i to be the safety score of class i, for each $1 \le i \le k$, and sort the k classes in increasing order of s_i.*

Let $Q_0 = (0, 0)$, $Q_{k+1} = (1, 0)$, and for each $1 \le i \le k$, define $Q_i = (s_i, v_i)$, where

$$v_i = \left(1 - \frac{1}{P}\sum_{j=1}^{i} p_j - \frac{1}{N}\sum_{j=1}^{i} n_j\right) s_i + \frac{1}{N}\sum_{j=1}^{i} n_j.$$

Then the cost curve corresponding to this model consists of $k + 2$ points, formed by connecting Q_i to Q_{i+1} for each $0 \le i \le k$.

Proof Let $s_0 = 0$ and $s_{k+1} = 1$. Since the classes are ranked in increasing order of s_i, for any fixed $x = PC(+)$, there is a unique index i (with $0 \le i \le k$) for which $s_i \le PC(+) < s_{i+1}$.

From Theorem 4.3, the optimal binary classifier is the one where classes $s_1, s_2, \dots, s_i$ are predicted positive, and all others are negative. By definition, this classifier has

$$TPR = \frac{1}{P}\sum_{j=1}^{i} p_j, \qquad FPR = \frac{1}{N}\sum_{j=1}^{i} n_j.$$

The point (FPR, TPR) in ROC space corresponds to the line $y = (1 - TPR - FPR)x + FPR$ in cost space, by Proposition 3.2. Given the value of x, we can determine the corresponding value of y.

If $x = s_i$, then $y = (1 - TPR - FPR)x + FPR = v_i$, as defined above. Therefore, each of the points $Q_i = (s_i, v_i)$ appear on the cost curve, for each index i.

Furthermore, the slope of an optimal cost line is constant for each $s_i \le x < s_{i+1}$. Thus, we can construct the cost curve directly from the $k + 2$ points $Q_0, Q_1, \dots, Q_{k+1}$, by joining each adjacent pair of points.

Since s_i and v_i are functions of p_i and n_i, we have proven that the cost curve can be generated directly from the predictive model, without having to draw cost lines or lower envelopes. This concludes the proof. □

Therefore, we can generate a cost curve directly from a classification model, using safety scores. By Lemma 4.4, we can determine these points explicitly and connect them to form the model's cost curve. We can take this one step further to determine a formula for the improvement function, using Corollary 4.2.

Theorem 4.5 *Let C_1 and C_2 be classification models, where C_1 is the trivial model that predicts every instance to be negative. For each $x = PC(+)$ in $[0, 1]$, let i be the unique index* *for which $s_i \leq x < s_{i+1}$, where the s_i's represent the safety scores of model C_2. Furthermore, define*

$$v_k = \left(1 - \frac{1}{P}\sum_{j=1}^{k} p_j - \frac{1}{N}\sum_{j=1}^{k} n_j\right) s_k + \frac{1}{N}\sum_{j=1}^{k} n_j,$$

for $k = i$ and $k = i + 1$. Then,

$$I_{[C_1,C_2]}(x) = 1 - \left[\frac{v_{i+1} - v_i}{s_{i+1} - s_i} \cdot \left(1 - \frac{s_i}{x}\right)\right] - \frac{v_i}{x}.$$

Proof For any $0 \leq x \leq 1$, Lemma 4.4 tells us how to compute the corresponding y-coordinate on the cost curve. First we construct the cost curve by joining (s_i, v_i) to (s_{i+1}, v_{i+1}) for each $i = 0, 1, \ldots, k$. For a fixed x, there is a unique index i for which $s_i \leq x < s_{i+1}$. It follows that the point (x, y) lies on the cost curve, where

$$y = \frac{v_{i+1} - v_i}{s_{i+1} - s_i}(x - s_i) + v_i.$$

By Corollary 4.2, $I_{[C_1,C_2]}(x) = 1 - \frac{y}{x}$. Substituting for y establishes the result. □

5 Marine container inspection

As described in Sect. 1, the Canada Border Services Agency (CBSA) is facing a very real problem with their marine container inspection processes, wasting unnecessary time and resources. In this section, we develop a simple classification model based on just four features to decide which containers should be ventilated without performing the initial chemical test. All containers that are not predicted to be positive retain the status quo, where a chemical test takes place initially to eliminate the risk of a customs officer walking into a toxic fumigated container.

We mentioned earlier that in this context, a false positive occurs when the classifier incorrectly predicts that a non-fumigated container has been fumigated, leading to an unnecessary ventilation. And a false negative occurs when the classifier fails to predict that a fumigated container has been fumigated, leading to an unnecessary initial chemical test. Thus, C^+ denotes the cost of a ventilation and C^- the cost of a chemical test. A perfect classifier will have no false negatives or false positives, and will be the optimal scenario for CBSA.

There are four variables to consider: the number of fumigated containers (P), the number of non-fumigated containers (N), the cost of a false positive (C^+), and the cost of a false negative (C^-). As mentioned in the Introduction, the goal is to develop a classifier that minimizes the total cost of misclassification, given by the formula

$$M = FN \cdot C^- + FP \cdot C^+.$$

In the context of improving efficiency at our marine ports, the "cost" could refer to time or money. Whether we are measuring cost in dollars or minutes, the misclassification cost is defined the same way: the goal is to minimize this quantity as much as possible. The strength of improvement curves is that we can perform the analysis without knowing the exact values of any of these unknown variables.

5.1 Data set and methodology

We were given a data set of 4,193 referred containers, of which 580 were fumigated (approximately 14%). For each container, only four features were provided:

(a) Origin country
(b) Canadian port of arrival
(c) HS section (e.g., Sect. 5 = Mineral Products)
(d) HS chapter (e.g., Chap. 26 = Ores, Slag, Ash)

Harmonized System (HS) codes are a government-regulated system used to classify products and their corresponding tariffs, and are used throughout the international shipping industry. There are 21 possible HS sections, which denote broad categories of what is being shipped. Within each section is a list of HS chapters, which provide more specific information on the type of commodity that is inside the marine container. In total, the 21 HS sections give rise to 99 different HS chapters.

As the intention of this paper is to give a specific application of improvement curves, we will only provide one predictive model. Of course, further analysis could be done to create the optimal model, and ensure its robustness by re-running the modeling process with hundreds of different training sets as is done in bootstrapping. But for the scope of this paper, we will provide just a simple classification model based on these four data elements. Despite having only four features, we can create a relatively strong predictive model (with ROC AUC = 0.75) that will be the basis for a highly reliable binary classifier.

We split the data set into a *training set* comprising of 70% of the data, and a *testing set* comprising of the remaining 30%. In our randomly-selected partition, the training set has 2,935 containers, of which 420 were fumigated (14.3% of instances are positive). And the testing set has 1,258 containers, of which 160 were fumigated (12.7% of instances are positive).

5.2 The predictive model

As mentioned, our classification model will be developed from four data elements. For simplicity, we use the following abbreviations: Cn = Country of Origin, Pr = Port of Arrival, Se = HS Section Number, and Ch = HS Chapter Number.

For each of these four features, we use the training set to tabulate the number of positive and negative containers for each possible value belonging to this feature. Suppose there are v_j possible values for feature j (for $j = 1, 2, 3, 4$). Then $v_3 = 21$ and $v_4 = 99$ since there are 21 possible values for Se and 99 for Ch.

For each feature j, we calculate the *support* and *confidence* for each of the v_j possible values. For example, if 7% of the containers in our training set arrive at the Port of Halifax, and of these containers, 10% are positive, then Halifax has a support of 7% and a confidence of 10%.

We designate a value of feature j as *high risk* (H) if its support is at least 2% and the confidence is at least 20%; otherwise denote it as *low risk* (L).

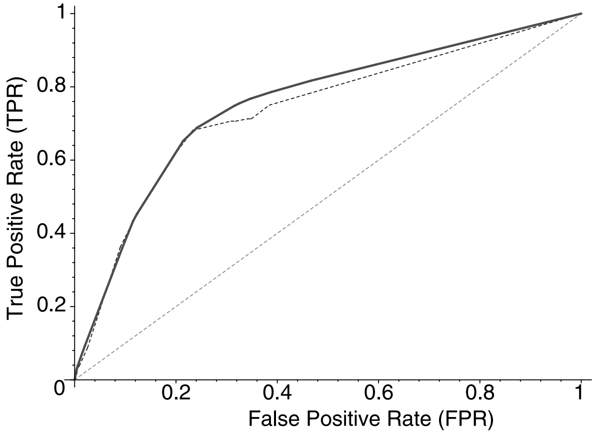
Hence, every Canadian port of arrival is designated either high risk or low risk. The same is true for origin countries, HS section numbers, and HS chapter numbers. (Note that the support and confidence thresholds of 2% and 20% were selected uniformly to create this model, but this is not a requirement.) Therefore, each container can be represented by a four-tuple such as $(Cn, Pr, Se, Ch) = (\text{L}, \text{H}, \text{L}, \text{L})$.

Our model will consist of 16 classes, corresponding to the $2^4 = 16$ possible four-tuples. For each of the sixteen classes (denoted by the indices $i = 1, 2, \ldots, 16$), we calculate the

Table 4 Sixteen classes with their safety scores

i	Cn	Pr	Se	Ch	p_i	n_i	s_i
1	H	H	H	L	12	10	0.12
2	L	H	H	L	1	1	0.14
3	H	L	L	H	3	4	0.18
4	H	H	L	H	23	36	0.21
5	H	H	H	H	112	185	0.22
6	L	H	H	H	31	54	0.23
7	L	H	L	H	8	19	0.28
8	H	H	L	L	84	230	0.31
9	L	H	L	L	15	68	0.43
10	L	L	H	H	25	184	0.55
11	L	L	H	L	3	26	0.59
12	L	L	L	H	5	51	0.63
13	H	L	H	L	1	13	0.68
14	H	L	H	H	7	94	0.69
15	H	L	L	L	13	197	0.72
16	L	L	L	L	77	1343	0.74

Fig. 5 ROC curves for model



safety score s_i based on the number of positive and negative containers (p_i and n_i) satisfying that class.

Note that all the calculations are performed on the 2,935 element training set with $P = 420$ and $N = 2515$. Table 4 provides the classification model, where the 16 classes are sorted by safety score (in increasing order). Once again, the s_i scores have been rounded to two decimal places.

As discussed in Sect. 4, sorting the classes by s_i score guarantees a convex ROC curve for the training set, but not necessarily for the testing set. Figure 5 illustrates the ROC curves for this classification model, for both the training set (thick solid line) and testing set (thin dotted line). The ROC AUC is 0.75 for the training set and 0.74 for the testing set. Notice how similar the curves are; in some regions, the two graphs essentially overlap.

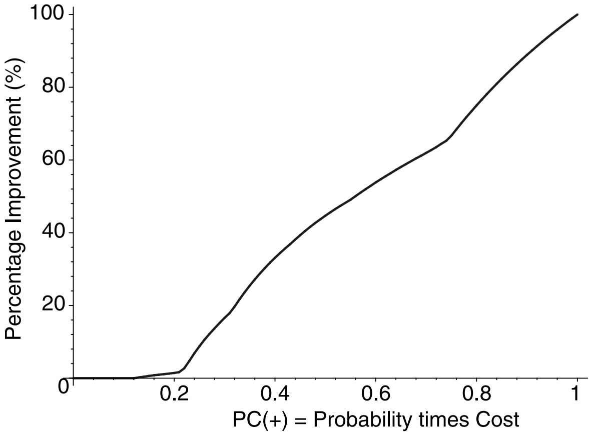


Fig. 6 Improvement curve for model

Thus, we have found a predictive model that is surprisingly robust, as their ROC curves are so similar. Certainly with a ROC AUC of 0.75, this is far from the optimal result, but this model is more than adequate to improve marine container risk-assessment at CBSA.

As demonstrated in Theorem 4.5, we can determine the improvement curve directly from the model's safety scores. We measure the percentage improvement of implementing this classification model over the status quo policy of performing an initial chemical test on every container (which is analogous to the predictive model where every instance is predicted to be negative). Figure 6 illustrates the improvement curve for this classification model, determined on the training set.

For each value of $x = PC(+)$, this graph measures the expected percentage improvement of replacing the status quo (of performing the initial chemical test on every container) with the model's best possible binary classifier.

Given this graph, how do we apply it? Specifically, we are interested in answering the following questions:

(1) How can a decision-maker interpret this graph and approve the use of a predictive model to replace the status quo policy of performing an initial chemical test on every container?
(2) How does the predictive model turn into a deployable binary classifier that would improve efficiency at our marine ports?
(3) Simulating this binary classifier on an independent data set, what would be the actual percentage improvement over the status quo?

We provide an example illustration to answer all three questions; the same methodology can be repeated for any specific scenario. Prior to providing this illustration, we explain the general methodology.

(a) First, we estimate the ratio $\frac{C^-}{C^+}$. This is the most important step as the expected percentage improvement will be off-target if this ratio is calculated incorrectly.
(b) Given the ratio $\frac{C^-}{C^+}$, determine t and x, using the values of P and N from the *training set*. Recall that $t = \frac{P \cdot C^-}{N \cdot C^+}$ and $x = \frac{P \cdot C^-}{P \cdot C^- + N \cdot C^+} = \frac{t}{t+1}$.
(c) Determine the expected percentage improvement by taking this value of x, and reading off the corresponding y-coordinate on the improvement curve.
(d) Produce the optimal binary classifier by setting this value of x as the safety score threshold, using Theorem 4.3.

(e) Simulate the results of this binary classifier by evaluating it on the *testing set* to measure the actual percentage improvement of the model had it been deployed.

This methodology answers all three questions posed above. To illustrate the procedures more explicitly, we provide the following example scenario.

5.3 An example scenario

Suppose that C^- is expensive while C^+ is relatively cheap. Specifically, we consider the scenario where $\frac{C^-}{C^+} = 4$. Suppose that this ratio has been determined after much analysis of ventilation and chemical testing costs, and that all senior decision-makers are in agreement with the validity of this ratio.

In practice, the estimates for C^- and C^+ are found by adding up the various costs involved: for a ventilation, the only real cost is the salary of a customs officer for that fixed period of time; for a chemical test, there are various equipment costs (e.g. Drager tubes, Tedlar bags), in addition to the salary of the customs officer. Preliminary analysis shows that C^- is expensive while C^+ is cheap, and the ratio of $\frac{C^-}{C^+} = 4$ is close to the actual figure.

Note that "cost" could refer to money or time. If we are dealing with cost as money, in this particular illustration, a chemical test is four times as expensive as a ventilation. If we are dealing with cost as time, a chemical test takes four times as long to complete as does a ventilation. By the definition of $x = PC(+)$, the actual values of C^+ and C^- do not matter in measuring percentage improvement; only their ratio is important.

From the training data, $P = 14.3\%$, so $\frac{P}{N} = \frac{14.3}{85.7} \sim \frac{1}{6}$. Therefore, $t = \frac{P \cdot C^-}{N \cdot C^+} \sim \frac{4}{6}$. It follows that $x = \frac{t}{t+1} \sim 0.4$. Performing the exact calculations, we determine that $x = 0.400$, rounded to three decimal places.

Now we determine the expected percentage improvement, by reading from the improvement curve. The x-coordinate of 0.4 corresponds to the y-coordinate of 33.09. Therefore, our expectation is an overall improvement of around 33%.

A decision-maker now has all the resources to answer Question (1). Just by reading the expected percentage improvement of 33% from the improvement curve, she can simply say, "This approach is definitely worth implementing. I approve it."

Now we answer Question (2). To turn the predictive model into a binary classifier, we use Theorem 4.3. Since $x = 0.4$, the desired threshold for the safety score is 0.4. Table 4 lists the $2^4 = 16$ classes of our classification model. From this table, only the first eight scenarios will be predicted positive as their s_i scores are less than or equal to 0.4. Summarizing, the binary classifier is illustrated in Table 5.

This model is easily deployable. All that is required is for a front-line border officer to have a chart with the list of high risk and low risk values for each of the four given features. When a container arrives at a particular port, the officer just needs to determine the class (or 4-tuple) for that container, and then read off Table 5 to determine the classifier's prediction.

If the prediction is positive, we ventilate the container without performing the initial chemical test. If the prediction is negative, then we follow the status quo and perform the chemical test first and then use the results of this initial test to determine whether the container requires ventilation. This answers Question (2).

We now answer Question (3) by measuring the actual improvement on independent data. We simulate the results of this binary classifier on our blind *testing set*.

From the data in the testing set, 12.7% of the containers are positive, which is slightly less than the fumigation rate of the training set used to create the model. We have $\frac{P}{N} = \frac{12.7}{87.3} = 0.145$, which implies that $t = \frac{P \cdot C^-}{N \cdot C^+} = 0.582$. It follows that $x = \frac{t}{t+1} = 0.368$, rounded to

Table 5 The optimal binary classifier for $x = 0.4$

i	*Cn*	*Pr*	*Se*	*Ch*	*Prediction*
1	H	H	H	L	Positive
2	L	H	H	L	Positive
3	H	L	L	H	Positive
4	H	H	L	H	Positive
5	H	H	H	H	Positive
6	L	H	H	H	Positive
7	L	H	L	H	Positive
8	H	H	L	L	Positive
9	L	H	L	L	Negative
10	L	L	H	H	Negative
11	L	L	H	L	Negative
12	L	L	L	H	Negative
13	H	L	H	L	Negative
14	H	L	H	H	Negative
15	H	L	L	L	Negative
16	L	L	L	L	Negative

Table 6 Confusion matrix for testing set

	Predicted P	Predicted N
Actual P	100	60
Actual N	223	875

three decimal places. From the improvement curve, it is anticipated that the actual improvement will be less than the classifier's expected value of 33%, since $x = 0.368$ is to the left of $x = 0.4$.

To determine the exact improvement, we calculate the classifier's *FPR* and *TPR* on the testing set. The confusion matrix is given in Table 6.

We have $TPR = \frac{100}{160} = 62.5\%$ and $FPR = \frac{223}{1098} = 20.3\%$. (As a side note, the *TPR* and *FPR* for the training set are very similar, at 65.2% and 21.4%, respectively).

In any cost curve, the normalized expected cost is given by the formula $y = f(x) = (FNR - FPR)x + FPR$. For the testing set, we have $FPR = 0.203$, $FNR = 1 - TPR = 0.375$ and $x = 0.368$. Substituting these values into the above formula, we determine that $y = f(x) = 0.266$. Compare this to the normalized expected cost of the status quo, which is simply $y = x = 0.368$. By Corollary 4.2, the actual improvement is

$$\text{Improvement} = \frac{x - f(x)}{x} = \frac{0.102}{0.368} \sim 28\%.$$

Thus, the results of the above analysis show clearly that we could improve upon current processes by deploying our simple model. Moreover, the output is presented in a way that is easily accessible to a non-mathematical audience. This answers Question (3).

Indeed, the actual 28% improvement is not too far off from the expected improvement of 33%. Of course, if we had more positive instances (i.e., the testing set had close to the 14.3% positive rate as in the training set), then the actual percentage improvement would have been much closer.

5.4 Results of analysis

We can repeat this process for each possible value of the ratio $\frac{C^-}{C^+}$, which in turn gives us all possible values of t and x. Following the methodology presented in the previous section, we can determine the expected and actual percentage improvement for any value of $x \in [0, 1]$.

This is indicated in Fig. 7, for both the expected improvement modeled on the training set (thick solid line) and the actual improvement evaluated on the testing set (thin dotted line).

From Fig. 7, we can determine the estimate and actual percentage improvement for any value of x. To illustrate, Table 7 provides these two measures for nine uniformly distributed values in this range. Each percentage has been rounded to the nearest integer.

As we saw in the case $x = 0.4$, the actual improvement is less than the expected improvement. Nevertheless, the actual percentage improvement is not as important as the fact that deploying this model would have saved both time and money. Certainly, if the analysis of estimating C^+ and C^- shows that $x \sim 0.4$, this simple 4-feature predictive model is definitely worth implementing: a decision-maker would be ecstatic with a 28% reduction in misclassification cost!

On the other hand, if the actual improvement was relatively minor (e.g., less than 5%), then the cost-benefit analysis would not justify deployment. From the figures in Table 7, we

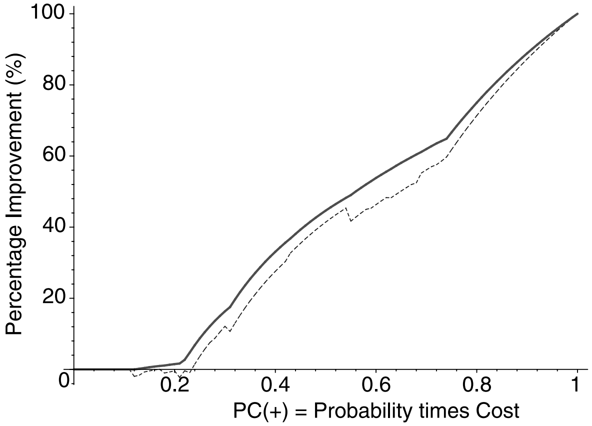


Fig. 7 Estimated and actual improvement

Table 7 Overall results for model

x value	Estimated improvement	Actual improvement
0.1	0%	0%
0.2	1%	0%
0.3	17%	12%
0.4	33%	28%
0.5	45%	41%
0.6	54%	46%
0.7	62%	56%
0.8	75%	71%
0.9	89%	87%

conclude that for this particular model, implementation makes sense provided that x is at least $0.25 \pm \varepsilon$. Of course, a better model would have a lower threshold.

The inequality $x \geq 0.25$ is equivalent to $t \geq \frac{1}{3}$, since $x = \frac{t}{t+1}$. By the definition of t, this inequality is equivalent to

$$\frac{P \cdot C^-}{N \cdot C^+} \geq \frac{1}{3}.$$

Initial analysis has been conducted to estimate the costs of C^- and C^+. Based on this analysis, we can justify to senior CBSA decision-makers that the above inequality holds, and hence a predictive model would be worth deploying. Especially as recent data has shown a notable increase in the percentage of positive fumigated containers, even a basic predictive model (such as our model with four features and a ROC AUC of 0.75) would lead to improved efficiency and cost savings at our marine ports.

Of course, if the percentage of fumigated containers decreases (i.e., the ratio $\frac{P}{N}$ becomes small) and new technology is developed to significantly reduce the costs of a chemical test (i.e., the ratio $\frac{C^-}{C^+}$ becomes small), then the inequality $\frac{P \cdot C^-}{N \cdot C^+} \geq \frac{1}{3}$ will no longer hold. In this case, the status quo would be superior to this classification model.

6 Conclusion

By introducing machine learning techniques to an operations research problem, we were able to develop a simple yet effective approach to improving efficiency at CBSA marine ports. Furthermore, the theories developed in this paper describe an easy way for decision-makers to estimate the potential improvement of introducing a predictive model to the overall risk assessment process, without having to determine the exact number of positive and negative containers, or knowing the exact costs of a ventilation or chemical test.

If new technology is developed to decrease the costs of a ventilation or chemical test, we can simply recalculate t and x and read off the potential benefit directly from the improvement curve. Knowing the expected percentage improvement would help decision-makers determine whether the status quo policy should be kept, or replaced by a binary classifier modeled on historical data. This approach of measuring expected percentage improvement via the Improvement Curve has been presented to senior managers at the Canada Border Services Agency, and has been acknowledged by the Agency's Chief Scientific Officer as "a much better way to visualize and evaluate performance than traditional approaches such as the ROC curve."

Of course, when a binary classification policy is approved, much work will be conducted to determine the best possible classifier for deployment. In addition to generating a robust predictive model with a high ROC AUC, we would also ensure that the classifier is monotonic. For example, the 4-tuple (H, L, H, H) is at least as risky as (H, L, L, H), by definition. However, in our model (see Table 5), the latter is predicted negative while the former is predicted positive.

In order to obtain buy-in from our border service officers, a monotonic classifier is essential. Also, the model would have to be simple and user-friendly to aid CBSA officers. While our four-feature classifier satisfies this criterion, further research would need to be conducted to determine the best possible approach that simultaneously maximizes reliability and robustness while retaining simplicity.

Acknowledgements The authors thank the referees for their insightful comments that significantly improved the content and presentation of this paper. We also acknowledge the editing suggestions of Sylvain Coulombe, Terri Webster, Dmitry Gorodnichy, and Huda Masoud.

References

Boström, H. (2005). Maximizing the area under the ROC curve using incremental reduced error pruning. In *Proceedings of the ICML 2005 workshop on ROC analysis in machine learning*.

Boström, H. (2007). Maximizing the area under the ROC curve with decision lists and rule sets. In *Proceedings of the SIAM international conference on data mining* (pp. 27–34).

Bradley, A. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, *30*(6), 1145–1159.

Drummond, C., & Holte, R. C. (2000a). Explicitly representing expected cost: an alternative to ROC representation. In *Proceedings of the fifth international conference on knowledge discovery and data mining* (pp. 155–164).

Drummond, C., & Holte, R. C. (2000b). Exploiting the cost (in)sensitivity of decision tree splitting criteria. In *Proceedings of the 17th international conference on machine learning* (pp. 239–246).

Drummond, C., & Holte, R. C. (2004). What ROC curves can't do (and cost curves can). In *ECAI workshop on ROC analysis in artificial intelligence*.

Drummond, C., & Holte, R. C. (2006). Cost curves: an improved method for visualizing classifier performance. *Machine Learning*, *65*, 95–130.

Fawcett, T. (2001). Using rule sets to maximize ROC performance. In *Proceedings of the 2001 IEEE international conference on data mining*.

Flach, P. A. (2003). The geometry of ROC space: understanding machine learning metrics through ROC isometrics. In *Proceedings of the twentieth international conference on machine learning* (pp. 194–201).

Flach, P. A. (2004). The many faces of ROC analysis in machine learning. In *Proceedings of the twenty-first international conference on machine learning*.

Furnkranz, J., & Flach, P. A. (2003). An analysis of rule evaluation metrics. In *Proceedings of the twentieth international conference on machine learning* (pp. 202–209).

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, *143*, 29–36.

Holte, R. C. (2006). Elaboration on two points raised in "Classifier technology and the illusion of progress". *Statistical Science*, *21*(1), 24–26.

Provost, F., & Fawcett, T. (1997). Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. In *Proceedings of the third international conference on knowledge discovery and data mining* (pp. 43–48).

Provost, F., & Fawcett, T. (2001). Robust classification for imprecise environments. *Machine Learning*, *42*, 203–231.

Swets, J. A. (1988). Measuring the accuracy of diagnostic systems. *Science*, *240*, 1285–1293.

Swets, J. A., Dawes, R. M., & Monahan, J. (2000). Psychological science can improve diagnostic decisions. *Psychological Science in the Public Interest*, *1*, 1–26.