Data Mining and Knowledge Discovery (2019) 33:805–847
https://doi.org/10.1007/s10618-019-00613-7

Setting decision thresholds when operating conditions are uncertain

Cèsar Ferri[1] · José Hernández-Orallo[1] · Peter Flach[2]

Received: 3 October 2017 / Accepted: 3 December 2018 / Published online: 16 February 2019

Abstract

The quality of the decisions made by a machine learning model depends on the data and the operating conditions during deployment. Often, operating conditions such as class distribution and misclassification costs have changed during the time since the model was trained and evaluated. When deploying a binary classifier that outputs scores, once we know the new class distribution and the new cost ratio between false positives and false negatives, there are several methods in the literature to help us choose an appropriate *threshold* for the classifier's scores. However, on many occasions, the information that we have about this operating condition is *uncertain*. Previous work has considered ranges or distributions of operating conditions during deployment, with expected costs being calculated for ranges or intervals, but still the decision for each point is made as if the operating condition were certain. The implications of this assumption have received limited attention: a threshold choice that is best suited without uncertainty may be suboptimal under uncertainty. In this paper we analyse the effect of operating condition uncertainty on the expected loss for different threshold choice methods, both theoretically and experimentally. We model uncertainty as a second conditional distribution over the actual operation condition and study it theoretically in such a way that minimum and maximum uncertainty are both seen as special cases of this general formulation. This is complemented by a thorough experimental analysis investigating how different learning algorithms behave for a range of datasets according to the threshold choice method and the uncertainty level.

Keywords Classification · Threshold choice methods · Uncertainty · Operating condition · Calibration

Responsible editor: Johannes Fürnkranz.

✉ Peter Flach
Peter.Flach@bristol.ac.uk

Extended author information available on the last page of the article

1 Introduction

It is now generally recognised in machine learning that optimal decisions depend on an appropriate identification and use of the operating condition surrounding the problem at hand. In classification, the operating condition is usually represented by the class distribution and the costs of misclassification. For instance, an undetected fault (false negative) in a production line can be far more critical than a false alarm (false positive) depending on the kind of product that is been manufactured. In this case, the kind of product, the deadline of the order and other factors determine the operating condition. While in general this operating condition can present itself in many ways, in important cases it can be integrated in the utility function or cost function. If a decision disregards the operating condition then it is very likely to be suboptimal. Conversely, if we predict the class by taking proper account of the operating condition, better decisions can be made.

But when is the operating condition known, and how much is effectively known about it? Sometimes we have this information when we train and test our models, and the operating condition does not change when the model is finally deployed to classify a new case. For instance, in the manufacturing example, if we train and test our models in a production line that always manufactures the same product and deploys the model in this very same line, the operating condition is well known and constant. This is represented by case A in Table 1. On other occasions, "the class distribution and costs can change with time, or [...] the distribution in the datasets used for training and evaluating the classifier may not reflect reality" (Drummond and Holte 2006). For the manufacturing line, this happens when deadlines or products change, but we have perfect information about them and the associated costs when the model is deployed. This is case B in the table. On the other extreme, one can also have a situation where the operating condition may be changing, but we do not know how it changes, as represented by case D. All these cases have been analysed in prior work and approaches to deal with them are readily available, as indicated in the right-most column of the table.

The present paper is about case C, a common situation situated between cases B and D: the operating condition has changed, and we have some (imprecise, uncertain) information about the most likely change. For example, we may have assumed equal misclassification costs when training the classifier, but now we need to handle a situation or operating condition c where false negatives are *approximately* twice as costly as false positives.

There are two ways of understanding and addressing case C. On the one hand, we can consider a range of operating conditions, e.g., an interval or a distribution, where c would be somewhat in the middle, and calculate the expected cost in this range of scenarios. This "precise information [being] unavailable when the system is run" (Provost and Fawcett 2001) has been considered in the literature under different terms and mechanisms: "imprecise distribution information [defining a] range of slopes" (Provost et al. 1997), "an interval within which he or she is confident" (Adams and Hand 1999) or "uncertain deployment scenarios" (Johnson et al. 2015). However, for each particular operating condition, all these approaches still calculate the threshold as if c *were* true. On the other hand, we can explicitly model that there is an uncertainty

Table 1 Four different cases according to changes in the operating condition (o.c.) during deployment and how much is known about this change

	Description	Approaches
A	Perfect knowledge about operating condition during deployment. The o.c. is the same as the one the model was trained with	Cost-sensitive learning, over- and under-sampling, robust methods to class imbalance, etc. (Elkan 2001)
B	Perfect knowledge about o.c. during deployment. The o.c. changes from the one the model was trained with	Threshold choice using ROC analysis, DEC curves or cost curves. Calibration (Fawcett 2006; Drummond and Holte 2006; Zadrozny and Elkan 2001b)
C	Uncertain knowledge about o.c. during deployment. The o.c. may change from the one the model was trained with, where some changes are more likely than others	This paper
D	No knowledge about o.c. during deployment. The o.c. can change from the one the model was trained with in any possible way	Aggregated metrics, such as AUC, Brier score, etc. (Hernández-Orallo et al. 2012; Flach et al. 2011; Zadrozny and Elkan 2001a; Liu et al. 2011)

around c and make the decision accordingly, which we can then aggregate or not in regions or intervals. In this paper, we address this scenario in a systematic way, modelling the uncertainty of the operating condition *explicitly* and analysing how to set a threshold optimally according to that uncertainty.

In classification, given case A in Table 1 we do not need the classifier to be very flexible, as we do not have changing conditions. A crisp classifier outputting class labels can suffice. For the other three cases, we need more versatile classifiers, whose predictions can be adjusted. In particular, when we talk about thresholds we are assuming that the models are able to estimate *scores* for the classes, and we compare these scores against the threshold, so making the predictions of one or the other class more likely. These scores usually represent some kind of confidence on the prediction, which, if well calibrated, matches the probability that an example is of a particular class. For example, in a well-calibrated binary classifier, if we consider the set of instances for which the model predicts scores around 0.8 then approximately 80% of these instances belong to the positive class.

In binary classification, given the *true* operating condition and a classifier outputting scores for the classes (e.g., probability estimations), we are in case B, for which there are several procedures in the literature to choose the threshold between positive and negative classification. These procedures are known as *threshold choice methods*. Previous work (Hernández-Orallo et al. 2011, 2012, 2013) has investigated these methods systematically, and the situations in which some are preferable over others. The analysis has also been extended to case D, by studying aggregated metrics of performance, such as the area under the ROC curve or the Brier Score, assuming a uniform distribution of operating conditions, considering this a very entropic situation modelling total uncertainty of the operating condition. However, for the common case C in Table 1, we do not know whether these results can be extrapolated—or interpolated between cases B and D. In other words, how do the threshold choice methods behave for case C? This is the main question addressed in this paper.

Using a model of uncertainty based on the Beta distribution, we provide a theoretical analysis, accompanied by graphical illustrations in terms of cost curves, as well as an extensive empirical evaluation, where several threshold choice methods are analysed for varying degrees of uncertainty. Our empirical results demonstrate different behaviours depending on the kind of dataset and the classification technique, with several factors affecting the performance, such as the calibration of the scores, the dataset imbalance, etc. Analysing these two parameters together (threshold choice method and operating condition uncertainty) for the first time suggests their addition as crucial variables in the learning process as well as in model selection procedures.

In summary, this paper makes a number of novel contributions. First, we consider that the operating condition can be known during deployment with degrees of uncertainty. Second, we model this by means of a cost parameter drawn from a Beta distribution, which is appropriate as the cost parameter is a number between 0 and 1. The location parameter of this Beta distribution is the expected value (as if there were no uncertainty) and a second parameter controls the shape and hence the uncertainty. Third, we perform a theoretical analysis of uncertainty in relation to several threshold choice methods. Fourth, we perform a complete experimental analysis about how different techniques behave for a range of datasets.

The paper is organised as follows. We first further motivate and illustrate case C in the next section. In Sect. 3 we review related work, focusing on the threshold choice methods used for performing model selection and configuration, and different kinds of uncertainty. Section 4 introduces basic notation, the formal definitions of operating conditions, cost curves and threshold choice methods. In Sect. 5 we formalise the notion of operating condition uncertainty and analyse how uncertainty affects threshold choice methods, deriving a series of theoretical results for cost curves and particularly for expected loss for complete uncertainty. Section 6 includes a thorough experimental evaluation with a range of learning techniques, datasets and threshold choice methods for different levels of uncertainty, and extracts a series of findings from the analysis. Finally, Sect. 7 closes the paper with an overall assessment of the results and some future work.

2 The case of operating condition uncertainty

As a motivating example, consider the following simple spam filtering scenario. Detecting spam in email, SMS, or other kinds of communications represents a paradigmatic use of machine learning (Guzella and Caminhas 2009). This is also a representative example to illustrate varying operating conditions, since the cost and prevalence of spam depends on the user: the amount of spam messages each user receives can vary considerably, and additionally the associated costs also differ widely over users as they depend on many factors (Ridzuan et al. 2010). Experiments in Sakkis et al. (2003) consider several operating conditions, some representing that

Table 2 Cost matrices for a spam filtering application

		Predicted	
		Spam	Ham
(a)			
Actual	Spam	–	\$0.28
	Ham	\$0.04	–
(b)			
Actual	Spam	–	\$0.21
	Ham	\$0.07	–

(a): Cost matrix as assumed when making a decision with a tentative cost proportion $\hat{c} = 0.28/(0.28 + 0.04) = 0.875$. (b): True cost matrix for the user with a true cost proportion $c = 0.21/(0.21 + 0.07) = 0.75$

blocking a good message is as bad as letting 99 spam messages pass the filter, but also operating conditions where blocking 1 good message has the same cost as letting 1 spam message in. In other words the cost ratios go from 99 to 1 depending on the user. Nevertheless, even if we know the exact preferences of a single user, the context of the problem is highly uncertain because of the varying proportions of the spam deliveries. For instance, Figures 1 and 3 in Fawcett (2003) report the percentage of spam over total e-mail for a population of users.[1]

Consider a previously learnt scoring classifier is used to determine whether to flag a message as spam or ham. Table 2a shows the *assumed* cost matrix to deploy the model for a particular user, which suggests that misclassifying a non-spam email for a given user is $0.28/0.04 = 7$ times more costly than misclassifying spam email. This can be integrated into an assumed *cost proportion* (i.e., the cost of a false negative in proportion to the joint cost of one false negative and one false positive) of $\hat{c} = 0.28/(0.28 + 0.04) = 0.875$. Now imagine that for a particular message m during this *deployment* stage the classifier outputs an estimated spam probability of 0.8. Using a *score-driven* threshold choice method—which compares the estimated probability with the assumed cost proportion $\hat{c}$—we would classify the message as ham, as this minimises the expected loss ($0.8 \cdot$ \$0.04 $< (1 - 0.8) \cdot$ \$0.28).

But because of the uncertainty of the problem, the costs were an assumption. The *true* costs may be somewhat different, as depicted in Table 2b. This is an example of case C in Table 1: it is not only that the costs have changed from training/test to deployment, but also that the assumed costs during deployment were wrong. The true operating condition is seen to be represented by a cost proportion $c = 0.75$. Using this true cost proportion, the optimal decision would be to classify message m as ham ($0.8 \cdot$ \$0.07 $> (1 - 0.8) \cdot$ \$0.21). We see that the assumed cost matrix leads to a suboptimal decision, *not because the classifier is poorly calibrated, but because the operating condition has been poorly estimated*. Note that it is not that we were completely wrong about the operating condition. It is 0.75 when we thought it was 0.875 so we are not in case D in Table 1.

[1] For the variability of global trends, one can check https://www.talosintelligence.com/reputation_center/email_rep or https://www.statista.com/statistics/420391/spam-email-traffic-share/.

What is of interest for our present investigation is that this uncertainty of the operating condition does not affect all threshold choice methods in the same way. In the example above, we used the score-driven threshold choice method, but other methods can be more sensitive to this uncertainty and magnify this error in the operating condition. Let us explore this with some real data. We trained a decision tree model (J48 Unpruned) for the Spambase dataset of the UCI repository (Lichman 2013). This dataset contains a collection of spam and valid emails, and it has been used to build spam filters. The model was trained with 99% of the data. Train dataset and test dataset have a similar class proportion. Figure 1 shows the performance on the remaining 1%, in the form of a cost curve. A cost curve (Drummond and Holte 2006) is a powerful tool to represent performance of binary classifiers on the full range of cost contexts. Cost lines in this plot show possible outputs of the performance of the model depending on the threshold employed.

There are different methods to set the threshold according to the cost context. The most common methods are train-optimal, estimating the optimal thresholds from the training dataset, with the cost curve in blue in Fig. 1, and score-driven, as described above, giving the red cost curve in Fig. 1. Test-optimal is represented in green. This method assumes to be able to choose the best threshold for each test instance, and hence gives an over-optimistic baseline (the lower envelope of all cost lines). These and other methods, such as rate-driven (using the intended predicted positive rate to derive the threshold), have been studied in previous work (Hernández-Orallo et al. 2011, 2012, 2013), showing that train-optimal may be affected by overfitting and score-driven by bad calibration, while rate-driven incurs a constant baseline cost. The use of cost curves allows us to express these different threshold choice methods in a common currency, expected loss, and decide which method to use depending on the situation. For instance, we see that for a cost proportion of $c = 0.15$ train-optimal has an expected loss of around 0.2 while test-optimal and score-driven have an expected loss of around 0.1. However, this analysis performed in previous work considers that the operating condition during deployment is perfect, i.e., they were considering case B in Table 1.

Consider now the same cost proportion for deployment as above, but now it is just an estimation $\hat{c} = 0.15$. Again, using score-driven or test-optimal, we would expect to have a loss of around 0.1. However, due to the high uncertainty in the context, imagine that the true cost context is actually $c = 0.4$. This means that we have to follow the cost line that grows steeply from (0,0), which is prolonged in dotted grey. The true loss we actually have with the chosen cost line goes from 0.1 to slightly above 0.3 as we see with the green arrows in Fig. 1. In this uncertain situation, train-optimal turns out to be a better choice, despite being theoretically a worse option because of its worse area under the curve. For $c = 0.15$, it selects the cost line between 0.26 and 0.16, but it even goes slightly down for the actual context of $c = 0.4$ (shown with the blue arrows). Note that taking a region or interval around $c = 0.4$ or $\hat{c} = 0.15$ and calculating an expected cost for that region (one traditional way of modelling uncertainty, as we discussed in the introduction) would never show that train-optimal is better sometimes, as in the case that we are analysing. In other words, there is no region where train-optimal is better than test-optimal, but we can still find decisions, under uncertainty, where train-optimal might be better.

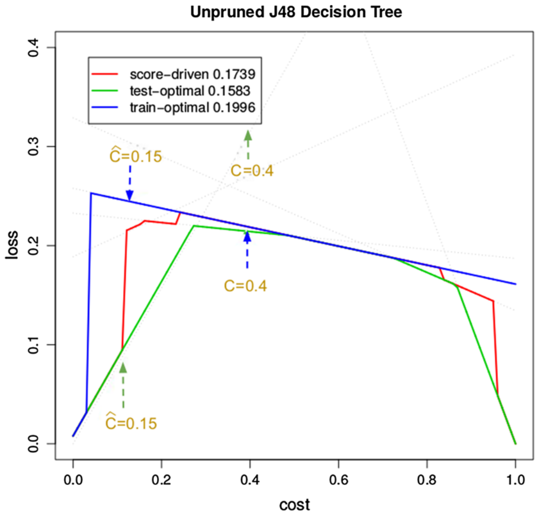


Fig. 1 Cost curve for an unpruned J48 tree and the spambase dataset. The x-axis shows the cost ratio c and the y-axis shows the expected loss. Three curves are shown corresponding to threshold choice methods score-driven, test-optimal and train-optimal. The area under these curves is included in the legend

In this paper we analyse this more realistic case when using several threshold choice methods, by taking into account that the assumed operating condition can be *uncertain*. For instance, if we use both train-optimal and score-driven threshold choice methods when we are given an operating condition of $\hat{c} = 0.2$, which method is more robust if this information has a high degree of uncertainty (and in the end it might well be $c = 0.1$ or $c = 0.5$)? How should we treat information about the operating condition that is vague or noisy?

The key issue to analyse this case C in Table 1 is to model this uncertainty. For a new user, a good approximation might be to take the operating conditions that we have seen for a sample of existing users, which is simply a distribution of cost ratios. We choose to model the uncertainty in $\hat{c}$ by a Beta distribution with mode c and *certainty* γ (we use an alternate parametrisation of the usual Beta(α, β) with $\alpha = c\gamma + 1$ and $\beta = (1 - c)\gamma + 1$). Hence certainty in $\hat{c}$ ranges from $\gamma = \infty$, i.e., complete certainty modelled by a delta function at $\hat{c} = c$ (case B); to $\gamma = 0$, i.e., complete uncertainty modelled by a uniform distribution over $\hat{c} \in [0, 1]$ (case D). Figure 2 includes the distribution of operating conditions considering two values of γ (10 and 100) and an estimated operating condition with mode in 0.2.

We analyse how expected loss and cost curves change for increasing uncertainty to the case of total uncertainty ($\gamma = 0$). We find that many threshold choice methods collapse and become equivalent to their uniform counterparts (e.g., score-driven becomes score-uniform) and we can analytically derive the cost curves and expected losses in this extreme situation of complete uncertainty. This allows us to frame the evolution from perfect operating condition information to total absence of information for all threshold choice methods, where case C in Table 1 becomes a generalisation of both cases B and D.

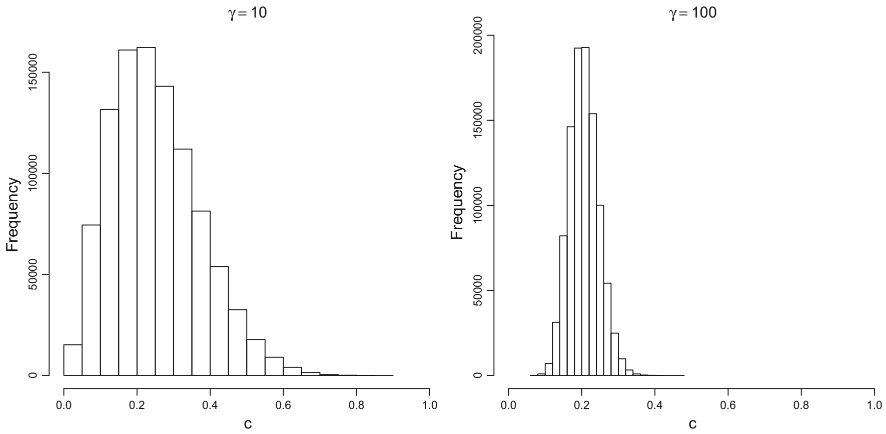


Fig. 2 Histograms (using 1,000,000 values) of $\hat{c}$ for a cost value $c = 0.2$ and two different levels of certainty: $\gamma = 10$ and $\gamma = 100$

3 Related work

Research on threshold choice methods mostly originates from the general area of decision making, ROC analysis and cost-sensitive learning in machine learning. A threshold choice method is a particular decision rule that converts the prediction or score provided by a model to a final decision. For instance, one of these methods is provided by ROC analysis, by calculating the Convex Hull of the ROC curve (Flach 2004; Fawcett 2006) [or, equivalently, applying the Pool Adjacent Violators algorithm (Fawcett and Niculescu-Mizil 2007)] to obtain the non-dominated operating points, each of which correspond to an optimal decision threshold for a range of class distributions and/or cost proportions. ROC analysis is usually performed on a training or validation dataset, which means that the decisions are optimal (in the sense of taking the most from the model) for that dataset. It is then important to distinguish between the term *train-optimal* to denote when the decisions are chosen optimally according to the training (or validation) dataset and the term *test-optimal*, which refers to the optimal decisions that would be made assuming the ROC curve could be estimated from the test data (which is usually impossible). Similarly, when we refer to calibration (Zadrozny and Elkan 2001b; Bella et al. 2013; Fawcett and Niculescu-Mizil 2007), it is important to clarify that a model is calibrated for a specific class distribution or dataset (usually the training or validation dataset). These distinctions are important because otherwise (for perfect ROC curves and perfect calibration) choosing according to the convex hull or choosing according to probability estimations would be equally optimal.

Accordingly, recent works have studied other threshold choice methods, such as the so-called *score-driven* and *rate-driven* methods, which might be more appropriate depending on the quality of the ranking and/or the probability estimates. This has led to a better understanding of aggregate performance metrics such as AUC or the Brier score (Hernández-Orallo et al. 2011, 2012, 2013), which are seen as expected loss for a particular threshold choice method for a range of operating conditions. Most of this work has been theoretical or oriented towards new types of visualisation (e.g., all threshold choice methods can be visualised on the same cost space, as all are expressed in terms of expected loss). An exception is de Melo et al. (2014), which considers cost and different threshold choice methods for proposing algorithm similarity measures from a meta-learning perspective. However, no full experimental analysis has been performed comparing the main realistic threshold choice methods (train-optimal, score-driven and rate-driven) with test-optimal.

The issues about the usual inductive extrapolation from training to test, and ultimately deployment are not the only cause of suboptimal decisions. Some of these other causes are usually grouped under the term ‘uncertainty’, a widely-researched topic in machine learning that has been covered in almost all aspects that can affect model selection and decision making. For instance, much research effort has focused on uncertainty in data, i.e., noise in the output and/or input features used to learn a model or the instances to be predicted (Bishop 2011). In this context, some works have developed versions of existing learning techniques such as decision trees (Tsang et al. 2011; Qin et al. 2009) or Naive Bayes (Ren et al. 2009) to handle this uncertainty. Other papers have addressed data uncertainty in a cost-sensitive setting (Liu et al. 2011; Huang 2015). In Dalton (2016), the authors address uncertainty from a different perspective, they assume the true population densities are members of an uncertainty class of distributions.

The uncertainty in the predictions produced by a classifier is also a very common topic in machine learning, and it has usually been linked to calibration (Zadrozny and Elkan 2001b; Bella et al. 2013; Fawcett and Niculescu-Mizil 2007). Calibration is closely related to the performance of several threshold choice methods, as it has been analysed theoretically (Hernández-Orallo et al. 2012), although not experimentally.

None of the above notions of uncertainty concern uncertainty or noise in the operating condition (case C in Table 1). However, it is well-recognised in machine learning that determining a precise cost matrix is problematic (Provost et al. 1997; Drummond and Holte 2006; Adams and Hand 1999; Provost and Fawcett 2001), as we have discussed in the introduction. Yet surprisingly, only a few works have focused on solving or systematically analysing this issue. Zadrozny and Elkan (2001a) is a seminal work on making optimal decisions when costs and probabilities are both unknown by introducing decision tree and Naive Bayes methods for obtaining well-calibrated probability estimates. A more recent approach is presented in Liu and Zhou (2010), where learning is done assuming cost intervals, specifically with support vector machines. They consider that the false positive and negative costs are taken as $c_0 = 1$ and c_1 ranging in a predefined interval $[c_{min}, c_{max}]$, which is given. They finally consider any possible distribution, showing that the expected value of the distribution is not the best decision in general, and thereby proposing a method which is based on sampling from the cost distribution, learning independent classifiers and then combining their outputs, which is essentially a Monte Carlo approach. In other words, the model assumes that we have access to the true distribution of c, and hence can sample many $\hat{c}$ from this distribution. However, this is unrealistic in many situations.

Similarly, Wang and Tang (2012) consider multiple cost matrices, and the goal is to minimise loss considering all of them are given, so there is no real uncertainty, either.

This is as if the models were evaluated for a set of operating conditions. Recently, Johnson et al. (2015) consider hypothetical deployment scenarios engendered by uncertain operating environments. The authors employ different loss functions to model uncertainty including a Beta distribution. The paper proposes a boosting-based method, RiskBoost, that reduces classifier risk when considering the Beta distribution, as suggested by Hand (2009). Finally, Dou et al. (2016) also consider a set of cost matrices and derives a method based on rough set theory such that "the smallest possible cost and the largest possible cost are calculated", which can be seen as an intermediate approach between the interval case and the multiple matrices case.

It is important to clarify that all the above approaches are given the parameters of the distribution or the set of cost matrices during deployment. Consequently, despite the name of some of these works, they view uncertainty as variability, and require information about this variability specified as *a range* of operating conditions [as done, for instance, in Hernández-Orallo et al. (2013)]. Once the range is defined as a distribution on c, the local decisions are made as if the operating condition were the true one, and then aggregated into regions. In other words, there is only one distribution over the true costs. In the current paper, in contrast, we consider that we get a value of $\hat{c}$ that is sampled (following one conditional distribution) from a true c (following *another* unconditional distribution), but the distribution is not known by the decision making system. This is a more realistic situation and more in accordance with the notion of uncertainty. Of course, in our theoretical analysis and the experiments we use some distributions to generate the values of $\hat{c}$, but the decision rule is not aware of these distributions.

Finally, all the above approaches in the literature have explicitly or implicitly assumed a kind of test-optimal threshold choice method. It is unclear whether the results would be better or worse for some other threshold choice methods. We study this theoretically for an uncertainty model and experimentally for a range of datasets.

4 Preliminaries

In this section we introduce basic notation and the key concepts of threshold choice methods, expected loss under a distribution of operating conditions, and cost curves. Most of this section follows (Hernández-Orallo et al. 2012).

A *classifier* is a function that maps instances x from an instance space X to classes y from an output space Y. For this paper we will assume binary classifiers, i.e., $Y = \{0, 1\}$. A *model* is a function $m : X \rightarrow \mathbb{R}$ that maps examples to scores on an unspecified scale. We use the convention that lower scores express a stronger belief that the instance is of class 0, which we label the positive class. A *probabilistic model* is a function $m : X \rightarrow [0, 1]$ that maps examples to estimates $\hat{p}(1|x)$ of the probability of example x to be of class 1. In order to make predictions in the Y domain, a model can be converted to a classifier by setting a decision threshold t on the scores. Given a predicted score $s = m(x)$, the instance x is classified in class 1 if $s > t$, and in class 0 otherwise.

For a given, unspecified model and population from which data are drawn, we denote the score density for class k by f_k and the cumulative distribution function by F_k. Thus, $F_0(t) = \int_{-\infty}^{t} f_0(s)ds = P(s \leq t|0)$ is the proportion of class 0 points correctly classified if the decision threshold is t, which is the sensitivity or true positive rate at t. Similarly, $F_1(t) = \int_{-\infty}^{t} f_1(s)ds = P(s \leq t|1)$ is the proportion of class 1 points incorrectly classified as 0 or the false positive rate at threshold t; $1 - F_1(t)$ is the true negative rate or specificity.

Given a dataset $D \subset \langle X, Y \rangle$ of size $n = |D|$, we denote by D_k the subset of examples in class $k \in \{0, 1\}$, and set $n_k = |D_k|$ and $\pi_k = n_k/n$. Clearly $\pi_0 + \pi_1 = 1$. We will use the term *class proportion* for π_0. Given a model and a threshold t, we denote by $R(t)$ the predicted positive rate, i.e., the proportion of examples that will be predicted positive if the threshold is set at t. This can also be expressed as $R(t) = \pi_0 F_0(t) + \pi_1 F_1(t)$.

A deployment context or *operating condition* is usually defined in practice by a misclassification cost function and a class distribution. In this paper we concentrate on misclassification costs as operating condition since the case of varying class distributions can be mapped back to a cost-sensitive scenario (Elkan 2001; Flach 2014).

Most approaches to cost-sensitive learning assume that the cost does not depend on the example but only on its class, and then cost matrices can specify that some misclassification costs are higher than others (Elkan 2001). Typically, the costs of correct classifications are assumed to be 0. In this way, for binary models we can describe the cost matrix by two values $c_k \geq 0$ with one or both being strictly greater than 0, representing the misclassification cost of an example of class k. Two examples were given in Fig. 2. These costs can be normalised by setting $b = c_0 + c_1$, (which is strictly greater than 0), and $c = c_0/b$; we will refer to c as the *cost proportion*. Under these assumptions, an operating condition can be defined as $\theta = \langle b, c, \pi_0 \rangle$. The space of operating conditions is denoted by Θ. Given a classifier characterised by its cumulative distribution functions F_0 and F_1, the loss for an operating condition θ using threshold t is defined as:

$$Q(t; \theta) \triangleq b\{c\pi_0(1 - F_0(t)) + (1 - c)\pi_1 F_1(t)\} \quad (1)$$

A threshold choice method is a key issue when applying a model under different operating conditions, defined as follows:

Definition 1 *Threshold choice method.* A threshold choice method is a function $T : \Theta \rightarrow \mathbb{R}$ that given an operating condition returns a decision threshold.

Given a threshold choice method T, the loss for a particular operating condition θ is given by $Q(T(\theta); \theta)$. When we are not given the operating condition until deployment, we should evaluate the loss over a distribution of possible (or likely) operating conditions $w(\theta)$, leading to the following general definition of expected loss:

$$L \triangleq \int_{\Theta} Q(T(\theta); \theta) w(\theta) d\theta \quad (2)$$

As already mentioned we will assume the variability of the operating condition to be fully captured by the cost parameters, so π_k are assumed to be constant. Furthermore, we assume that b and c are independent and that thresholds are chosen solely in terms of c. Under these assumptions Eq. 1 is simplified as follows:[2]

$$Q(t; c) \triangleq \mathbb{E}\{b\}\{c\pi_0(1 - F_0(t)) + (1 - c)\pi_1 F_1(t)\} \tag{3}$$

The expected value of b appears in the loss expression as a scaling factor, and can conveniently be set to 2 in order for loss to be commensurate with error rate when costs are balanced ($c = 0.5$).

The expected loss for costs is then adapted from Eq. 2 as follows:

Definition 2 Given a threshold choice method T and a probability density function over cost proportions w, *expected loss* L_δ is defined as

$$L_\delta \triangleq \int_0^1 Q(T(c); c) w(c) dc \tag{4}$$

Here we use the symbol δ to denote absolute certainty about the operating condition (a Dirac delta distribution, as explained later).

If we plot $Q(T(c); c)$ against c we obtain *cost curves* as defined by Drummond and Holte (2000, 2006). For fixed thresholds these curves are straight lines running from $Q(t; 0) = 2\pi_1 F_1(t)$ for $c = 0$ to $Q(t; 1) = 2\pi_0(1 - F_0(t))$ for $c = 1$. Given n examples with different scores (no ties), assuming they are ordered by increasing score, changing a threshold anywhere between two consecutive scores does not change the decision. Consequently, these straight lines corresponding to these $n + 1$ cases are usually referred to as "cost lines", and are represented by the pair $Q(t; 0) \to Q(t; 1)$ of values at $c = 0$ and $c = 1$ respectively, or simply by their index i, between 0 to n.

We now properly introduce the threshold choice methods studied in this paper. Some theoretical properties of these methods can be found in Hernández-Orallo et al. (2012).

The first threshold choice method is based on the optimistic assumption that at deployment time we select the threshold that minimises the loss using the current model. This threshold choice method, denoted by T^o, is defined as follows:

Definition 3 The optimal threshold choice method is defined as:

$$T^o(c) \triangleq \arg\min_t \{Q(t; c)\} \tag{5}$$

$$= \arg\min_t \; 2\{c\pi_0(1 - F_0(t)) + (1 - c)\pi_1 F_1(t)\} \tag{6}$$

Note that the arg min will typically give a range (interval) of values which give the same optimal value. So this method could be called non-deterministic.

This threshold choice method was employed by Drummond and Holte (2000, 2006) to define their cost curves. The cost curve formed by applying this optimal threshold choice method corresponds to the lower envelope of the set of all cost lines, and according to Hernández-Orallo et al. (2012) the area under this lower envelope is related to

[2] Hernández-Orallo et al. (2012) considered a second version of the loss expression where the class prior is absorbed into a different operating condition (z). Here we stick to the above version.

the model's refinement loss, actually RL_{Conv}, the refinement loss after convexification of the model [convex hull calibration, (Flach and Matsubara 2007)].

From a practical point of view, we need to differentiate between the test and train (or validation) situations, as already illustrated above. The original optimal threshold choice method is applied over the same data that is used to compute the loss $\arg\min_t \{Q(t; c)\}$. This is clearly unrealistic in practical scenarios and is included here as an optimistic baseline to compare against; it will be called the test-optimal threshold choice method (sometimes just abbreviated to optimal), as we can usually see this with *test* data, but this is usually considered as an ideal situation during *deployment*. More realistically, the train-optimal threshold choice method $T^o(c)$ uses one set of data to set the threshold (usually training data) and another set of data to evaluate the associated loss.

The next threshold choice method treats the classifier's scores as calibrated probability estimates.

Definition 4 Assuming the model's scores are expressed on a probability scale [0, 1], the *score-driven threshold choice method* is defined as follows:

$$T^{sd}(c) \triangleq c$$

The curve for a given classifier defined as a plot of loss against operating condition using the score-driven threshold choice method is defined as the *Brier curve* (Hernández-Orallo et al. 2011). The area under the Brier curve is equal to the Brier score (Brier 1950). These curves are also useful to know the calibration loss of classifiers by comparing the difference of its Brier curve with respect to the test optimal cost curve.

The next method only considers the order of the scores and not its magnitude.

Definition 5 The *rate-driven threshold choice method* is defined as

$$T^{rd}(c) \triangleq R^{-1}(c)$$

where $R(t) = \pi_0 F_0(t) + \pi_1 F_1(t)$ is the predicted positive rate at threshold t or, equivalently, the cumulative distribution function of the classifier's scores. R^{-1} is hence the quantile function of the scores.

The *rate-driven cost curve* is defined as a plot of $Q(T^{rd}(c); c)$ and the area under this curve is directly related to AUC (Hernández-Orallo et al. 2013), exactly as $\frac{1}{3}+\pi_1\pi_0(1-2AUC)$. Unlike the other curves the rate-driven cost curve is *interpolating* between different cost lines, as some predicted rates may only be achievable by introducing fractional instances.

5 Uncertainty in the operating conditions

Consider again expected loss as in Definition 2. In this definition the operating condition c plays a dual role: as input to the threshold choice method $T(c) = t$, and as input to the loss function $Q(t; c)$, which calculates the loss at that threshold. However, in practice we do not have access to the exact operating condition when setting the threshold, but at best to an approximate, uncertain or noisy $\hat{c}$. The following definition of expected loss captures this more realistic operating condition uncertainty:

Definition 6 Given a threshold choice method for cost proportions T, a probability density function over *true* cost proportions w and a conditional density function v over *presumed* cost proportions given the true cost proportion, *expected loss* L_v is defined as

$$L_v \triangleq \int_0^1 K_v(c)w(c)dc \tag{7}$$

$$K_v(c) \triangleq \int_0^1 Q(T(\hat{c}); c)v(\hat{c}|c)d\hat{c} \tag{8}$$

In words, $K_v(c)$ is the expected loss at true cost proportion c, where the expectation is taken over the conditional density $v(\hat{c}|c)$; and L_v is $K_v(c)$ averaged over all c, where the expectation is taken over the density $w(c)$.

In real situations we do not know v exactly but we can assume a family of distributions that is sufficiently general and flexible to consider both the expectation and the uncertainty of the operating condition. Since costs are bounded between 0 and 1, a natural choice for $v(\hat{c}|c)$ is the Beta distribution $\text{Beta}(\alpha, \beta)$, which has two positive shape parameters α and β. It is furthermore natural to set the mode of this Beta distribution ($(\alpha - 1)/(\alpha + \beta - 2)$) equal to the expected cost proportion c as a location parameter. The remaining degree of freedom is fixed by choosing a *certainty level* $\gamma = \alpha + \beta - 2$ as a shape parameter, ranging from $\gamma = 0$ (complete uncertainty) to $\gamma = \infty$ (complete certainty). We can recover the conventional parameters as $\alpha = c\gamma + 1$ and $\beta = (1 - c)\gamma + 1$.

We define $K_{v[\gamma=x]}(c)$ as the expected loss at true cost proportion c, where $v(\hat{c}|c)$ is modeled by a Beta distribution using the γ parameter defined above fixed to x. Trivially, for $\gamma = \infty$ we have $c = \hat{c}$, so the Beta distribution becomes a Dirac delta distribution with all the mass in c and hence $K_{v[\gamma=\infty]}(c) = K_\delta(c) = Q(T(c); c)$ and $L_{v[\gamma=\infty]} = L_\delta$. The beta distribution ranges from this delta distribution to the other extreme, when $\gamma = 0$, which is a uniform distribution expressing complete uncertainty about the true operating condition.

We continue with an analytical analysis of the impact of this uncertainty model on expected loss under the various threshold choice methods. In what follows, we study how cost curves are affected by uncertainty. Then, we centre the analysis on the extreme case of complete uncertainy ($\gamma = 0$), first for non-interpolating threshold choice methods, and then for the (interpolating) rate-driven threshold choice method.

5.1 Expected loss as a weighted average of cost lines

We can decompose cost curves $Q(T(c); c)$ as follows.

Definition 7 Given $n + 1$ cost lines indexed by $i \in \{0, 1, \ldots, n\}$ (for instance, corresponding to a dataset with n examples without ties in the classifier scores), and T a non-interpolating threshold choice method, let us consider the portion of $Q(T(c); c)$ that involves cost line i. We define c_i (c_{i+1}) as the lowest (highest) value of c for which T selects cost line i. Clearly, $c_0 = 0$. If a cost line is never selected by T then $c_i = c_{i+1}$. The height of cost line i at c is denoted as $h_i(c)$. The trapezoid where a cost line i is used is then given by c_i, c_{i+1}, $h_i(c_i)$ and $h_{i+1}(c_{i+1})$.

As introduced above, $v(\hat{c}|c)$ is a distribution quantifying the uncertainty of $\hat{c}$; we denote its cumulative distribution by V. As cost lines are straight, the following lemma transforms the segment-wise view of $Q(T(\hat{c}); c)$ into a weighted aggregate of all cost lines that might be selected by $\hat{c}$.

Lemma 1 *Given $n + 1$ cost lines, the expected loss at point c is given by a weighted sum of the height of all $n + 1$ cost lines at c:*

$$Q(c) = \sum_{i=0}^{n} \Delta V_i(c) h_i(c)$$

where

$$\Delta V_i(c) = V(c_{i+1}|c) - V(c_i|c)$$

The proof of all theorems, lemmas and corollaries are found in Appendix A. This lemma expresses that under uncertainty, the expected loss at true cost proportion c is a 'vertical' average of all losses incurred by the different cost lines that may be selected through the noisy $\hat{c}$, weighted by the cumulative probability of obtaining a $\hat{c}$ that would select that cost line.

From this decomposition in weighted aggregates it is easy to see that the area of a curve can be computed as follows:

Theorem 1 *The area under a cost curve (expected loss) can be decomposed as:*

$$L_v = \sum_{i=0}^{n} L_i$$

where

$$L_i = \int_0^1 [V(c_{i+1}|c) - V(c_i|c)] h_i(c) dc$$

quantifies the contribution of a single cost line to the expected loss.

This theorem states how we can compute expected loss for the full range of c under an uncertainty function v by aggregating the contribution of the cost lines $Q(c)$ for the full range of c.

From the general expression we can derive the special cases for complete certainty and complete uncertainty. We need some notation first. For each cost line i we define $l_i = h_i(0)$ and $k_i = h_i(1) - l_i$, which means that the cost line is defined by $h_i(c) = k_i c + l_i$.

Theorem 2 *The contribution of cost line i to the expected loss for complete uncertainty is* $L_i = (c_{i+1} - c_i)\{k_i/2 + l_i\}$.

Theorem 3 *The contribution of cost line i to the expected loss for complete certainty is* $L_i = (c_{i+1} - c_i)\{k_i(c_{i+1} + c_i)/2 + l_i\}$.

While mathematically straightforward, Theorems 2 and 3 allow us to say more precisely under what circumstances complete certainty gives a lower expected loss for a given cost line than complete uncertainty: this happens exactly when

$$k_i(c_{i+1} + c_i) < k_i$$

For ascending cost lines with positive slope k_i this means $c_{i+1} + c_i < 1$: i.e., the midpoint of the segment must be below 1/2. Analogously, for descending cost lines the midpoint of the segment should be above 1/2. Taken together, the midpoint of the segment should be within the lower half of the cost line. This is desirable in general but cannot always be guaranteed for every cost line. However, we demonstrate in the next section that when aggregating over all cost lines, expected loss for complete uncertainty is indeed higher than for complete certainty for several threshold choice methods.

5.2 Expected loss under complete uncertainty

In this section we analytically derive (expected) cost curves and expected loss for several threshold choice methods under complete uncertainty (i.e., the conditional density of $\hat{c}$ given c is a uniform distribution in [0, 1], modelled by a Beta distribution with certainty parameter $\gamma = 0$).

5.2.1 Score-driven thresholds

Theorem 4 *Assuming scores in* [0, 1], *the expected cost curve for the score-driven threshold choice method and complete operating condition uncertainty is given by:*

$$K^{sd}_{v[\gamma=0]}(c) = 2\{\pi_1(1 - \overline{s}_1) + (\pi_0\overline{s}_0 - \pi_1(1 - \overline{s}_1))c\} \tag{9}$$

where $\overline{s}_k$ *represents the average scores for class k.*

This cost curve is hence a straight line, moving from $K = 2\pi_1(1 - \overline{s}_1)$ for $c = 0$ to $K = 2\pi_0\overline{s}_0$ for $c = 1$. For calibrated classifiers it is horizontal and independent of c, since in that case $\pi_0\overline{s}_0 = \pi_1(1 - \overline{s}_1)$ (Hernández-Orallo et al. 2013, Theorem 41).

Corollary 1 *The expected loss for the score-driven threshold choice method, uniform c and complete operating condition uncertainty is the classifier's mean absolute error:*

$$L^{sd}_{v[\gamma=0]} = \pi_0 \bar{s}_0 + \pi_1(1 - \bar{s}_1) = MAE$$

This result should be contrasted with score-driven thresholds under complete certainty of the operating condition, which leads to an expected loss equal to the Brier score (aka mean squared error). For probabilities squared error is never larger than absolute error, and hence the increase of expected loss from *BS* to *MAE* characterises precisely the additional loss incurred as a result of uncertainty in c. We will see later that this increase is maximally a factor 2 (for perfectly calibrated models).

These results also establish a connection with the *score-uniform* threshold choice method defined by Hernández-Orallo et al. (2013), which sets a random threshold on the scores irrespective of the operating condition c. They derive the same cost curve and expected loss as above (Hernández-Orallo et al. 2013, Theorem 11 and Corollary 12). In other words, setting the threshold according to an operating condition that is maximally uncertain is equivalent to ignoring the operating condition and setting the threshold randomly.

5.2.2 Rate-driven thresholds

Theorem 5 *The expected cost curve for the rate-driven threshold choice method and complete operating condition uncertainty is given by:*

$$K^{rd}_{v[\gamma=0]}(c) = 2\pi_0\pi_1(1 - AUC) + \pi_1^2 + (\pi_0 - \pi_1)c \tag{10}$$

We again get a straight cost curve, moving from $K = 2\pi_0\pi_1(1-AUC)+\pi_1^2$ for $c = 0$ to $K = 2\pi_0\pi_1(1-AUC) + \pi_1^2 + (\pi_0 - \pi_1) = 2\pi_0\pi_1(1-AUC) + \pi_0^2$ for $c = 1$. For balanced classes this line is horizontal and the loss independent of c.

Corollary 2 *The expected loss for the rate-driven threshold choice method, uniform c and complete operating condition uncertainty is related to the classifier's AUC as follows:*

$$L^{rd}_{v[\gamma=0]} = \pi_0\pi_1(1 - 2AUC) + 1/2$$

which is $1/6$ *higher than if the operating condition were fully known.*

We find that setting the threshold according to an operating condition that is maximally uncertain is equivalent to ignoring the operating condition and setting the threshold randomly, which is another result of Hernández-Orallo (2013, Theorem 18).

5.2.3 Optimal thresholds

Theorem 6 *The expected cost curve for the optimal threshold choice method and complete operating condition uncertainty is:*

$$K^{o}_{v[\gamma=0]}(c) = 2\pi_0\bar{s}^*_0$$

where s^* denotes scores obtained after perfect calibration. Hence the loss is independent of the true operating condition c.

Theorem 7 *The expected loss for the optimal threshold choice method with complete uncertainty is twice the expected loss for the optimal threshold choice method with complete certainty :*

$$L^o_{v[\gamma=0]} = 2L^o_\delta \tag{11}$$

Again, we have a result where complete uncertainty increases the loss with respect to complete certainty, this time very considerably by a factor two.

5.3 Discussion

Table 3 summarises our results so far for complete operating condition uncertainty. The first column gives the expected cost lines in the form $K(c) = A + Bc$, and the right column gives the expected loss (area under the expected cost line). Table 4 gives the corresponding results for complete operating condition certainty from Hernández-Orallo et al. (2013).

We have seen that the impact of uncertainty is different for different threshold choice methods:

- for score-driven thresholds the loss evolves from *BS* to *MAE* and hence the additional loss due to complete operating condition uncertainty depends on the model's performance;
- for rate-driven thresholds the additional loss is an additive constant (1/6);
- for (test-)optimal thresholds the additional loss is a multiplicative factor (2).

Table 3 Expected cost curve and expected loss for several threshold choice methods, under complete operating condition uncertainty

	Expected cost curve $K_{v[\gamma=0]}(c)$	Expected loss $L_{v[\gamma=0]}$
Score-driven	$2\{\pi_1(1-\bar{s}_1) + (\pi_0\bar{s}_0 - \pi_1(1-\bar{s}_1))c\}$	$\pi_0\bar{s}_0 + \pi_1(1-\bar{s}_1) = MAE$
Rate-driven	$2\pi_0\pi_1(1-AUC) + \pi_1^2 + (\pi_0 - \pi_1)c$	$\pi_0\pi_1(1-2AUC) + 1/2$
Optimal	$2\pi_0\bar{s}^*_0$	$2\pi_0\bar{s}^*_0 = 2RL(\text{Cal(m)})$

Table 4 Corresponding results from Hernández-Orallo et al. (2013), for complete operating condition certainty

	Cost curve $K_\delta(c)$	Expected loss L_δ
Score-driven	$2\{c\pi_0(1-F_0(c)) + (1-c)\pi_1 F_1(c)\}$	*BS*
Rate-driven	$2\{c\pi_0(1-F_0(R^{-1}(c))) + (1-c)\pi_1 F_1(R^{-1}(c))\}$	$\pi_0\pi_1(1-2AUC) + 1/3$
Optimal	$2\{c\pi_0(1-F^*_0(c)) + (1-c)\pi_1 F^*_1(c)\}$	$\pi_0\bar{s}^*_0 = RL(\text{Cal(m)})$

$F^*_k(c)$ indicates the optimal F_k for that value of c

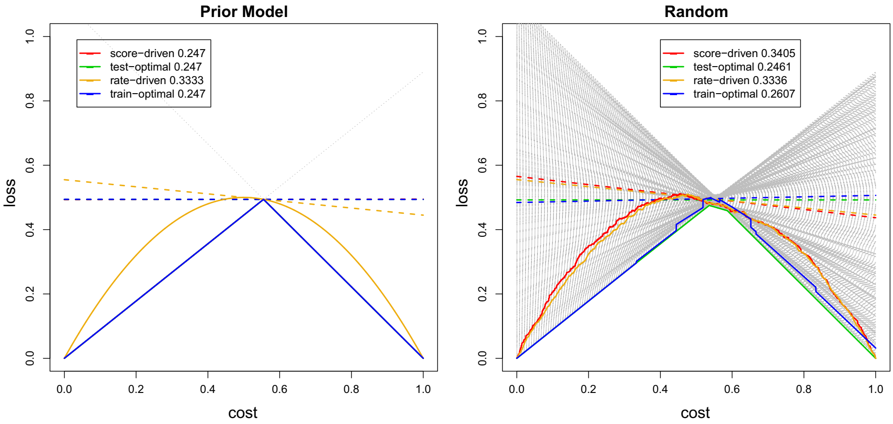


Fig. 3 Cost curves for the credit-a dataset for different threshold choice methods. The class proportion is $\pi_0 = 0.44$ and $\pi_1 = 0.56$. Left: Prior model, Right: Random model. Solid lines correspond to complete certainty ($\gamma = \infty$) and dashed lines correspond to complete uncertainty ($\gamma = 0$)

Hence it is conceivable that threshold choice method T_1 outperforms T_2 for one certainty level while the situation is reversed for another certainty level.

To illustrate, we consider two simple baseline models: a model predicting π_1 for all instances (henceforth called the prior model) and a random model assigning uniform random scores in [0, 1]. Figure 3 presents the results of these two basic models for the credit-a dataset from UCI (Lichman 2013). However, for these simple models we can easily derive the cost curves analytically:

Prior model: (Figure 3 (Left)) This model is calibrated by construction ($\overline{s}_0 = \overline{s}_1 = \pi_1$ and hence $\pi_0 \overline{s}_0 = \pi_1(1-\overline{s}_1)$), and so the score-driven thresholds are optimal. As the model is constant we also have that train and test optimal coincide. Their complete certainty curve consists of the two default cost lines $0 \rightarrow 2\pi_1$ and $2\pi_0 \rightarrow 0$, crossing at ($c = \pi_0$, $Q = 2\pi_0\pi_1$); the curve has an area of $BS = \pi_0\pi_1$. In the case of complete uncertainty this doubles to $MAE = 2\pi_0\pi_1$, giving the dashed horizontal line. The prior model has $AUC = 1/2$ and hence an expected rate-driven loss under complete certainty of $1/3$; the rate-driven cost curve is the same as for the random model (see below). The expected loss increases to $1/2$ for complete uncertainty, corresponding to a cost line from $(0, \pi_0)$ to $(1, \pi_1)$ (which is slightly tilted for credit-a as this dataset is not quite balanced).

Random model: (Figure 3 (Right)) A model that generates scores uniformly randomly between 0 and 1 has $F_0(s) = F_1(s) = R(s) = s$ (in expectation) and hence both the score-driven and rate-driven cost curves for complete operating condition certainty can be derived from Table 4 as $K_\delta(c) = 2c(1-c)$ with associated expected loss of $1/3$. As before, under complete uncertainty the expected loss increases to $1/2$ with an expected cost line from $(0, \pi_0)$ to $(1, \pi_1)$. Furthermore, train-optimal coincides with test-optimal (in expectation) so their complete certainty cost curves are composed of the default cost lines while the complete uncertainty line is again horizontal at loss $MAE = 2\pi_0\pi_1$.

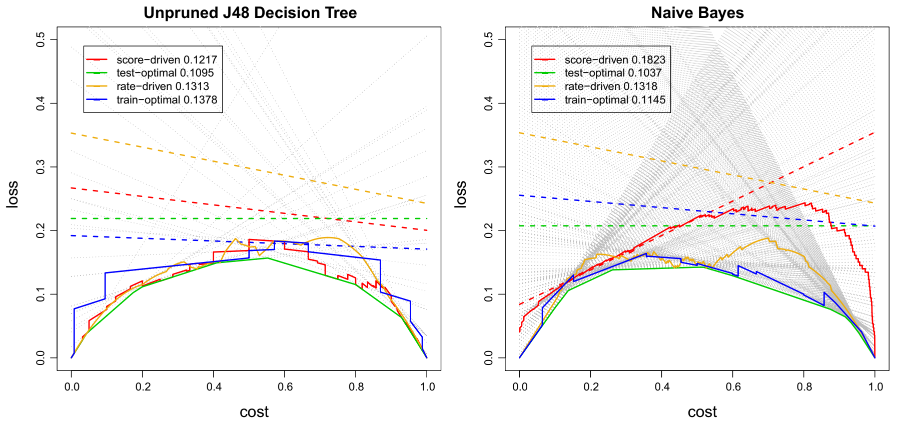


Fig. 4 More cost curves for the credit-a dataset. Different cost curves obtained by four different threshold selection methods (solid lines). Dashed lines correspond to complete uncertainty for credit-a dataset (50% train 50% test). Left, J48 decision tree unpruned with Laplace Correction. Right, Naive Bayes

If we move beyond the prior model and the random model, we get more elaborate curves for the two extremes. Figure 4 includes the curves for no uncertainty (solid lines) and the curves for complete uncertainty (dashed lines) for other models. The left figure shows the curves for the J48 decision tree unpruned with Laplace Correction from the RWeka package (Hornik et al. 2009). The right figure includes the curves for Naive Bayes. In these cases, we see that the curves without uncertainty dominate the curves with complete uncertainty for most of the range between 0 and 1, although there are a few small regions where the solid curve is above the dashed curve (e.g., train-optimal around 0.6 for J48 and score-driven between 0.15 and 0.5 approx. for Naive Bayes).

Please refer to Appendix B for a detailed example of using different threshold choice methods with complete certainty and complete uncertainty.

6 Experiments

In this section we will explore a series of questions experimentally, in order to shed more light on the effect of operating condition uncertainty in classification models. Specifically, we aim to investigate the following questions:

- Are the best configurations for complete certainty also the best when uncertainty increases?
- Is this persistent across model classes or does it depend on the way different models generate scores?
- Is it possible to recommend appropriate threshold choice methods according to the model type and an estimate of the uncertainty level?

6.1 Materials and methods

In order to do investigate the above questions, we vary on four dimensions:

Data: 26 datasets varying considerably in size (less than 100 to nearly 5000 instances; 4–168 attributes), class imbalance (relative size of the majority class from 0.50 to 0.97) and difficulty. They are shown in Table 5.

Classification Models: 12 representative model types including logical (decision trees/ lists/ stumps), geometric (support vector machines), distance-based (nearest neighbour), probabilistic (naive Bayes, logistic regression) and baseline (Prior, Random). See Table 6.

Threshold choice methods: 4 threshold choice methods: test-optimal, train-optimal, score-driven and rate-driven.

Uncertainty degree: We study the effect of uncertainty in $\hat{c}$ by means of different values of γ using the beta model.

To the best of our knowledge, this is the first time that these four characteristics are studied together experimentally. Even for the complete certainty cases, whereas there are several theoretical results about threshold choice methods in the literature, there has been no experimental analysis to date regarding the performance of these methods for different model types and datasets.

For simplicity and space, in this section we will summarise the results for the whole range between $c = 0$ and $c = 1$ by using the expected loss, i.e., we use the area under the cost curve as a performance measure, and we will only show a selection of cost plots. The code and data used for these experiments can be found in the repository Ferri et al. (2017).

For each dataset, we split the data in 50% for train and 50% test maintaining the class proportion in the datasets (stratified samples). We repeat this procedure 10 times. In this way, we learn $26 \times 12 \times 10 = 3120$ models. For each model, we compute the expected loss considering the four threshold choice methods (train-optimal, test-optimal, rate-driven and score-driven) over the test set.

We cover the range of the operating conditions (c) by performing 1001 steps with true values $c = 0, 0.001, \ldots, 0.999, 1$. In order to simulate uncertainty we use the true c as the mode of a Beta distribution with certainty level γ, as explained in Sect. 5. In that way, given a γ and a c value we generate values for $\hat{c}$, which are the presumed operating conditions that are used by the threshold choice methods. We use the values $\gamma \in \{\infty, 16, 8, 4, 2, 1, 0\}$. The true c is employed for computing the cost lines, the cost plots and the expected cost.

6.2 Comparison of threshold choice methods without uncertainty

We first analyse the case without uncertainty. To the best of our knowledge this is the first published experimental comprehensive comparison for a range of threshold choice methods, so we will first analyse this setting for its own sake and then we will compare this with the situation with uncertainty. Table 7 gives the performance obtained for different model types and threshold choice methods without incorporating operating condition uncertainty.

Table 5 Characteristics of the datasets used in the experiments

	Dataset	# Instances	# Attributes	Majority
1	Badges2	294	11	0.71
2	Breast-cancer	286	10	0.70
3	Breast-w	699	10	0.66
4	Bupa	345	7	0.58
5	Chess-KRVKP	3196	37	0.52
6	Credit-a	690	16	0.56
7	Credit-g	1000	21	0.70
8	Cylinder-bands	540	40	0.58
9	Diabetes	768	9	0.65
10	Haberman	306	4	0.74
11	Heart-statlog	270	14	0.56
12	Hepatitis	155	20	0.79
13	Ionosphere	351	35	0.64
14	Liver-disorders	345	7	0.58
15	Lung-cancer	32	57	0.72
16	Monks1	556	7	0.50
17	Monks2	601	7	0.66
18	Monks3	554	7	0.52
19	Musk1	476	168	0.57
20	Ozone	2536	73	0.97
21	Sonar	208	61	0.53
22	Spambase	4601	58	0.61
23	Spect_test	187	23	0.92
24	Spect_train	80	23	0.50
25	Tic-tac-toe	958	10	0.65
26	Vote	435	17	0.61

We use 26 binary datasets from the UCI repository (Lichman 2013). Here we include the size of the dataset (number of instances and attributes) as well as the proportion of the majority class

Specifically, for each model type we include the average Area Under the ROC Curve (*AUC*), the Brier Score (*BS*) and its decomposition in Calibration Loss (*CL*) and Refinement Loss (*RL*). This decomposition is computed using the method defined in Flach and Matsubara (2007). Given that performance metrics over different datasets are not necessarily commensurate, in the last two columns we include two pairwise comparisons between threshold methods: sc_vs_tr represents the proportion of cases that the score-driven method obtains a better or equal performance than train-optimal, while rd_vs_tr represents the proportion of cases that the rate-driven method obtains a better or equal performance than train-optimal.

If we analyse the results comparing the threshold choice methods, and regarding pairwise comparisons, we see that in general score-driven is able to get better or equal results compared to train-optimal in more than half of the cases. Excluding the

Table 6 Model types used in the experiments

Model type	Description
IBK1	k-Nearest Neighbours ($k = 1$)
IBK10	k-Nearest Neighbours ($k = 10$)
J48	J48 decision tree with pruning
J48Unp	J48 decision tree without pruning but with Laplace correction
Logist	Logistic Regression
NB	Naive Bayes
PART	Part decision lists
Prior	Returns π_1 for all instances
Random	Returns a uniformly random score in the interval [0, 1]
Stump	Decision stump
SVMp	Support vector machines computing output probabilities
SVM	Support vector machines outputting crisp scores (0 or 1)

All of them are included in the RWeka package (Hornik et al. 2009) and used with default parameters (unless stated otherwise), except "Prior" and "Random", which were implemented by the authors

baselines (Prior and Random), score-driven is better for all methods except for SVM and NB. These two methods have, on average, poor performance on *BS*. In fact, if we study the correlation of the difference of the rate-driven and train-optimal expected losses (not reported here), we find that there is a correlation of 0.44 of this difference with respect to *BS*, and 0.59 with respect to *CL*. This means that for models with high *BS* (hence poor calibration) train-optimal is the best choice, but in general *score-driven thresholds are competitive in the absence of operating condition uncertainty*. Nevertheless, we have to be careful, because even if the means are not commensurate, they are generally in favour of train-optimal (except for J48Unp). About rate-driven, notice that it often achieves the worst performance because this method employs all cost lines (even if some of them are never optimal), but in some cases (NB, IBK1) the difference with score-driven is relatively small.

To complete the picture for the uncertainty-free case, in Table 8 we include the results segmented by dataset and averaged by model type (except the baselines). This different aggregation of results is advantageous because for each row as the area under the cost curves for the four threshold choice methods are for the same dataset and this is a common currency (expected loss) and hence commensurate. This means that the means are now more meaningful for comparison than before. The entries are sorted in increasing expected loss under the test-optimal threshold, which establishes an optimistic baseline and indicates that the datasets indeed represent a wide range of difficulties (the worst possible test-optimal loss is 0.25—achieved by both baselines—which we get close to on the bottom three datasets: liver-disorders, bupa and cylinder-bands). In the last two columns we again include the relative pairwise comparison between score-driven vs. train-optimal and rate-driven vs. train-optimal, respectively. If we look at the means, train-optimal is advantageous over score-driven (except for dataset 3 and 11, and very slightly). The pairwise comparison gives more proportions in favour of score-driven, but the clearest wins are for train-optimal. Consequently,

Table 7 Results by model type averaged over all datasets, without uncertainty in the operating condition

Model	*AUC*	*BS*	*CL*	*RL*	Test optimal	Train optimal	Scoren driven	Tate driven	sc_vs_tr	rd_vs_tr
IBK1	0.751	0.200	0.062	0.138	0.138	**0.190**	0.200	0.221	*0.562*	***0.404***
IBK10	0.810	0.145	0.042	0.103	0.125	**0.143**	0.145	0.198	***0.485***	***0.135***
J48	0.750	0.149	0.028	0.121	0.125	**0.148**	0.149	0.216	*0.685*	***0.138***
J48Unp	0.798	0.142	0.030	0.112	0.119	0.159	**0.142**	0.201	*0.808*	***0.312***
Logist	0.804	0.160	0.145	0.014	0.115	**0.153**	0.160	0.194	*0.562*	***0.150***
NB	0.811	0.183	0.178	0.005	0.122	**0.145**	0.183	0.198	***0.131***	***0.092***
PART	0.768	0.154	0.032	0.122	0.125	**0.152**	0.154	0.211	*0.696*	***0.165***
Prior	0.500	0.216	0.000	0.216	0.216	**0.216**	**0.216**	0.333	*0.538*	***0.000***
Random	0.498	0.335	0.335	0.000	0.209	**0.224**	0.335	0.335	***0.008***	***0.008***
Stump	0.705	0.169	0.009	0.160	0.161	**0.168**	0.169	0.245	*0.731*	***0.008***
SVMp	0.818	0.138	0.127	0.011	0.112	**0.134**	0.138	0.197	*0.604*	***0.108***
SVM	0.735	0.188	0.055	0.133	0.133	**0.141**	0.188	0.224	***0.085***	***0.050***

The first four columns show performance in terms of four metrics. The following four columns indicate the area under the cost curve obtained with four threshold choice methods (note that the score-driven area equals the Brier score). In boldface the best results from the three realistic threshold choice methods: train-optimal, score-driven and rate-driven. The two last columns indicate pairwise comparisons: sc_vs_tr gives the proportion of cases that the score-driven method obtains better or equal performance than train-optimal, and rd_vs_tr does the same for rate-driven against train-optimal. Italics is used when proportion > 0.5 (bold italic when < 0.5)

Table 8 Results by dataset averaged over all model types, without uncertainty in the operating condition

dat.	*AUC*	*BS*	*CL*	*RL*	Test optimal	Train optimal	score driven	rate driven	sc_vs_tr	rd_vs_tr
1	0.999	0.002	0.002	0.000	0.001	**0.001**	0.002	0.127	*0.710*	***0.000***
20	0.692	0.059	0.040	0.019	0.025	**0.032**	0.059	0.314	***0.220***	***0.000***
3	0.969	0.041	0.013	0.028	0.036	0.042	**0.041**	0.118	***0.490***	***0.000***
26	0.964	0.057	0.028	0.029	0.045	**0.053**	0.057	0.115	***0.280***	***0.000***
5	0.946	0.057	0.021	0.037	0.052	**0.054**	0.057	0.112	***0.490***	***0.000***
23	0.688	0.095	0.047	0.049	0.062	**0.085**	0.095	0.304	*0.550*	***0.000***
25	0.899	0.092	0.036	0.056	0.081	**0.090**	0.092	0.153	*0.630*	***0.030***
22	0.917	0.102	0.037	0.064	0.084	**0.089**	0.102	0.140	*0.510*	***0.000***
18	0.901	0.108	0.048	0.059	0.087	**0.095**	0.108	0.133	***0.320***	***0.060***
19	0.889	0.117	0.044	0.073	0.092	**0.106**	0.117	0.149	*0.660*	***0.070***
13	0.877	0.121	0.047	0.074	0.093	**0.110**	0.121	0.161	*0.680*	***0.030***
6	0.868	0.144	0.059	0.085	0.122	**0.138**	0.144	0.152	***0.390***	***0.160***
12	0.683	0.185	0.081	0.104	0.134	**0.166**	0.185	0.267	***0.210***	***0.000***
16	0.808	0.166	0.080	0.085	0.138	**0.148**	0.166	0.179	*0.560*	***0.160***
15	0.669	0.228	0.111	0.117	0.146	**0.228**	**0.228**	0.262	***0.400***	***0.240***
11	0.828	0.172	0.062	0.110	0.145	0.175	0.172	**0.171**	*0.730*	***0.460***
9	0.762	0.197	0.073	0.123	0.171	**0.193**	0.197	0.214	*0.800*	***0.150***
7	0.702	0.218	0.091	0.127	0.179	**0.218**	**0.218**	0.249	*0.770*	***0.220***
10	0.596	0.221	0.092	0.129	0.181	**0.217**	0.221	0.296	*0.630*	***0.060***

Table 8 continued

dat.	*AUC*	*BS*	*CL*	*RL*	Test optimal	Train optimal	score driven	rate driven	sc_vs_tr	rd_vs_tr
21	0.753	0.247	0.108	0.139	0.181	0.240	0.247	**0.208**	*0.760*	*0.800*
2	0.628	0.233	0.114	0.120	0.187	**0.226**	0.233	0.280	***0.390***	***0.050***
24	0.701	0.271	0.126	0.145	0.195	0.267	0.271	**0.233**	*0.630*	*0.760*
17	0.570	0.247	0.114	0.134	0.211	**0.239**	0.247	0.302	*0.610*	***0.220***
14	0.617	0.281	0.122	0.158	0.223	**0.266**	0.280	0.277	*0.680*	***0.270***
4	0.617	0.281	0.122	0.158	0.223	**0.266**	0.280	0.277	*0.680*	***0.270***
8	0.608	0.291	0.124	0.167	0.225	**0.241**	0.291	0.282	***0.120***	***0.050***

The columns are the same as in Table 7. The rows are sorted on increasing test-optimal loss (harder data sets are further down the table). In boldface the best results from the three realistic threshold choice methods: train-optimal, score-driven and rate-driven. For the pairwise comparisons Italics is used when proportion $>$ 0.5 (bold italic when $<$ 0.5) on average for all model types, train-optimal seems a better option than score-driven. Only if we are sure which model type we are using, and we expect this to be well calibrated, should we use score-driven as a first option. Finally, about rate-driven, it seems to perform better against train-optimal for the harder datasets further down in the table, but the pattern is not sufficiently clear to recommend rate-driven consistently in any situation.

6.3 Robustness against uncertainty

In the previous section we compared different threshold choice methods with full certainty regarding the operating condition, which in the beta model is represented by $\gamma = \infty$. We now study the influence of decreasing operating condition certainty ($\gamma \in \{16, 8, 4, 2, 1, 0\}$). This is depicted in Fig. 5 for the same ten model types evaluated previously, for train-optimal and score-driven thresholds, averaged over all datasets. We also include the $\gamma = \infty$ results from Table 7 (columns trainoptimal and scoredriven).

The way uncertainty affects different model types for train-optimal thresholds is fairly gradual and similar, with one or two exceptions. Notice in particular that all curves are monotonic. There are few changes in the performance ranking of the model types until we reach high levels of uncertainty. For complete uncertainty, J48-Unpruned and IBK1 obtain the best performance, while they are among the three worst for total certainty. In other words, the relative increase in expected loss for high levels of uncertainty is lower for those two methods than for the others. For J48-Unpruned this is a consequence of the tree's low-coverage leaves, which leads to extreme probability estimates (slightly softened by the use of Laplace smoothing). A similar explanation goes for IBK1, which in fact only outputs crisp probabilities (0 or 1) and hence results in a three-point ROC curve with a single non-trivial operating point. The only reason why the curve is not completely flat is that the train-optimal threshold choice method can benefit from choosing one of the trivial operating points for extreme values of $\hat{c}$. Overall, from this plot, we can conclude that most model types can deal correctly with low levels of operating condition uncertainty with train-optimal thresholds.

In the case of score-driven thresholds we reach similar conclusions: uncertainty affects different model types gradually, similarly and monotonically. We see that the expected loss curve for IBK1 is completely flat, as score-driven thresholds will always choose the non-trivial operating point regardless of the operating condition. We see the same behaviour for SVM, at least if we use the version that outputs crisp probabilities. If we convert the SVM scores to probabilities by means of Platt scaling the behaviour changes considerably and is much more in line with the other model types. The difference is quite striking (SVM is the best method for complete uncertainty but the second-worst method for complete certainty) and relevant for machine learning practitioners, who may not always realise which version of the SVM model type they are actually using.

We continue by looking in more detail at the experimental results for complete certainty ($\gamma = \infty$) and complete uncertainty ($\gamma = 0$). We do this through a scatter plot, so we avoid averaging over datasets. For each of the 26 datasets we run 10 trials with different random train-test splits. Hence each point in the plots of Fig. 6

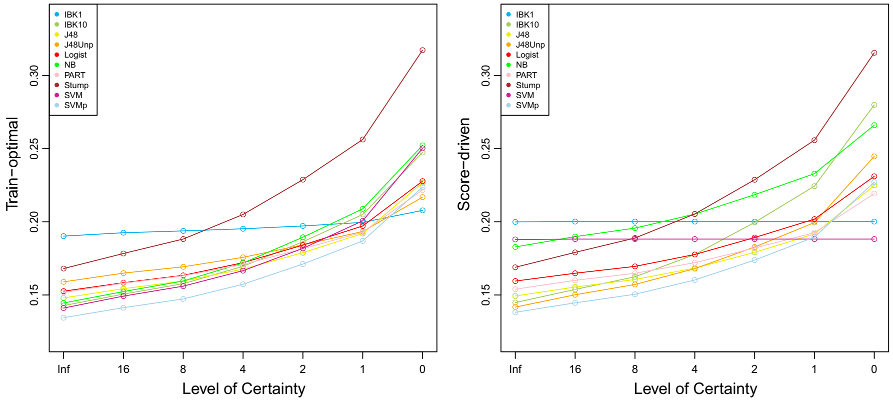


Fig. 5 Evolution of expected loss (area under the curve) with different levels of certainty ($\gamma \in \{\infty, 16, 8, 4, 2, 1, 0\}$) for different model types and threshold choice methods averaging all the datasets. Left: train-optimal. Right: score-driven

represents the expected loss of two threshold choice methods, score-driven on the y-axis against train-optimal on the x-axis for one of these trials and a particular model type (indicated by colour). We analyse this without uncertainty (top, $\gamma = \infty$) and total uncertainty (bottom, $\gamma = 0$). We include lines that represent the averages of every model type (distinguished by colour), horizontal lines are for expected loss computed by score-driven while vertical lines represent averages of expected loss with train-optimal (hence these averages can be traced back to the extreme points of the curves in Fig. 5). Points below the diagonal represents trials where score-driven outperforms train-optimal, while points above the diagonal indicate the opposite.

If we observe the plot without uncertainty (top, $\gamma = \infty$) we see some distinct patterns. The preference for train-optimal of SVM and NB is visible above the diagonal, indicating that their scores are poorly calibrated as probability estimates. Conversely, J48-Unpruned can be mostly found below the diagonal, indicating that train-optimal thresholds have a tendency to overfit. We also indicate the two baseline methods: Prior, returning the prior probability for class 1 (i.e., π_1); and Random, returning a uniform random score in [0, 1]. As derived earlier at the end of Sect. 5.3, Random has expected loss $\pi_0\pi_1$ for both score-driven and train-optimal thresholds under complete operating condition certainty (slightly under $1/4$ when averaged over the 26 datasets, some of which are unbalanced); Prior has the same expected loss for train-optimal thresholds but a fixed loss of $1/3$ for score-driven thresholds.

When we analyse the plot with complete uncertainty (Fig. 6, bottom) we find that, as expected, there is a considerable degradation in the performance of the models. This can be related back to our analytical results summarised in Tables 3 and 4: for score-driven thresholds the expected loss increases from *BS* to *MAE*, which can in the worst case represent a doubling (although the effect decreases when probabilities become more extreme); and for optimal thresholds the expected loss doubles (although this holds analytically only for test-optimal thresholds and may be softened by train-optimal thresholds). The plot shows that this generally degrades train-optimal thresholds more than score-driven ones (most obviously for SVM).

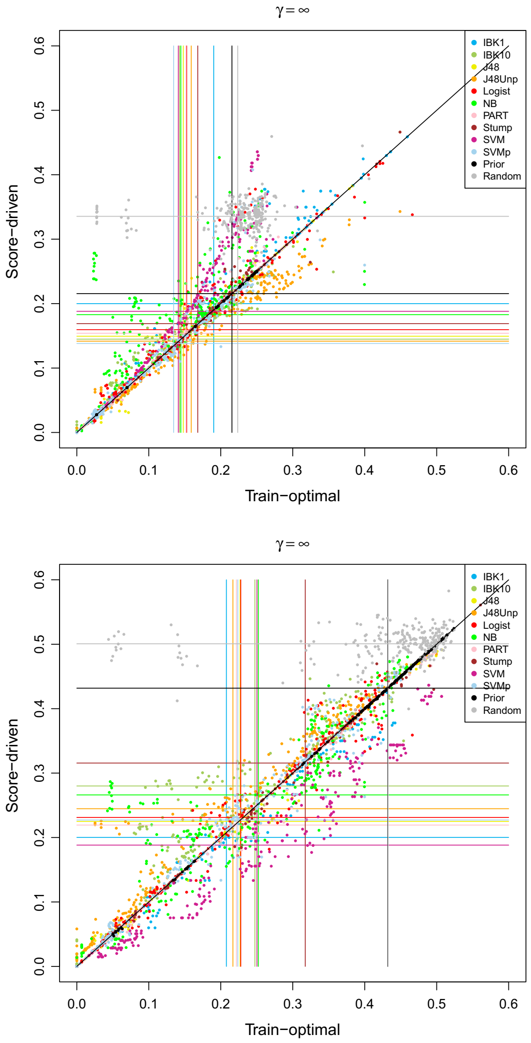


Fig. 6 Comparison between the expected losses for score-driven and train-optimal thresholds, without uncertainty (top) and with complete uncertainty (bottom). Model types are distinguished by colours, and for each model type we plot 260 points (10 train-test splits for each of the 26 datasets). Horizontal and vertical lines represent averages. Above (below) the diagonal score-driven performs worse (better) than train-optimal

Cost curves for some of the model types are given in Fig. 7. Here we can see in detail some of the behaviours detected in Figs. 5 and 6. When we compare the plots of J48 with pruning and without it, we can see how the pruning technique reduces the number of leaves in the tree and this is represented by the lower number of cost lines in the cost plot of J48 with pruning. The performance of all threshold choice methods is reduced in the unpruned version, however score-driven is not so affected. In fact, for the unpruned version score-driven obtains better performance than train-optimal. A similar behaviour is found when we compare IBK10 (using ten neighbours) versus IBK1 (using one neighbour). In the case of support vector machines, we can see the difference between the version that computes scores (SVMp) and the version without scores (SVM). In the case of the version without scores, score-driven selects

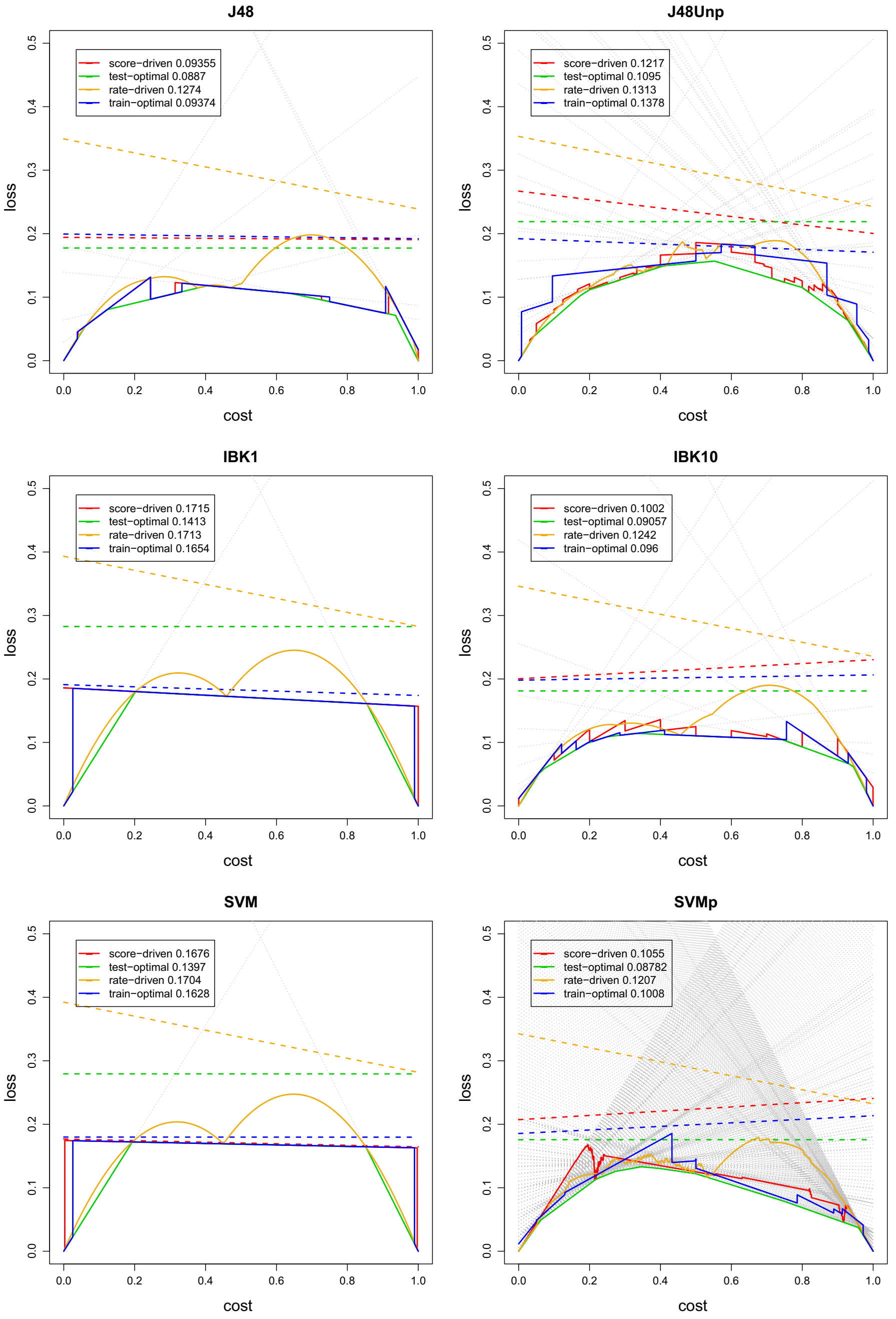


Fig. 7 Cost curves for selected models on the credit-a dataset. The legend includes the expected loss of the several threshold choice methods for complete certainty, and the dashed lines represent the expected curves for complete uncertainty

the same cost line regardless of the operating condition, and therefore it is invariant under uncertainty.

After the experiments, what can we say about the original questions at the beginning of this section? We can summarise our findings as follows:

- The best threshold choice methods for all degrees of certainty seem to be train-optimal and score-driven. Rate-driven thresholds give poor results in general, even for poorly calibrated models.
- Those model types that are good without uncertainty are usually good for slight degrees of uncertainty. However, for extreme uncertainty, those model types that generate extreme probabilities can get superior results for train-optimal and especially for score-driven thresholds.
- For no uncertainty, several metrics, such as the calibration loss component of the Brier score, might help estimate the performance for score-driven. This was known theoretically, but we have seen this experimentally in Tables 7 and 8.
- For high uncertainty, however, this connection was not that clear theoretically. In Tables 3 and 4 the change from BS to MAE for score-driven might be more noticeable for uncalibrated models but this is more independent of calibration for train-optimal (in the ideal case, the loss is just multiplied by 2). Consistently, now we see experimentally that there is no clear pattern between calibration and performance of score-driven for total uncertainty. Actually, for some poorly calibrated models the results were better than other better calibrated models, suggesting that if uncertainty is maximal, it might be better to turn the probabilities into crisp 0/1.

Overall, the effects are different and more diverse as the degree of uncertainty increases, which reinforces the hypothesis that the expected degree of uncertainty must be taken into account when selecting model type and threshold choice method.

7 Conclusion

The relevance of threshold choice methods to understand how classifiers behave is increasingly being appreciated. Classifiers cannot properly be evaluated if we do not specify what decision rule or threshold choice method will be used. Previous work has analysed the expected loss for a range of operating conditions. However, this previous work was done at the theoretical level for three threshold choice methods (optimal, score-driven and rate-driven) assuming that the given operating condition c is perfect. In this work, we first clarify that the optimal threshold choice method has usually been considered in an idealistic way—as if a ROC curve and its convex hull could alway be estimated perfectly—instead of a more realistic *train*-optimal threshold choice method. Secondly, we have considered uncertainty in the operating condition—as estimating misclassification costs precisely is problematic at best in many cases—and we have provided a theoretical analysis of what this implies for threshold choice methods. Thirdly, we have analysed the behaviour of these threshold choice methods without uncertainty and with complete uncertainty (and situations in between) experimentally, using a wide range of techniques and datasets.

In the case of no uncertainty, the results partly follow the intuitions from the theoretical results. For instance, the rate-driven threshold choice method has a constant term that makes it worse than the rest in almost all situations. However we also saw that the train-optimal threshold choice method often performs sub-optimally and hence has to be distinguished from the idealistic test-optimal threshold choice method. Broadly speaking, the train-optimal and score-driven threshold choice methods obtain similar performances, although there are differences depending on the model type and dataset that is employed. Here, we want to emphasise the advantage of score-driven threshold choice method since it can be applied directly without any other information or process, provided we have some reasonably well-calibrated scores. In contrast, the train-optimal threshold choice method requires to analyse which thresholds are optimal on a train dataset (for instance, through ROC analysis) and then those selections are applied for predicting new data. Obviously, performance of the train-optimal threshold choice method depends crucially on how well the train-optimal thresholds generalise to the test set.

When considering uncertainty of operating conditions, we proposed to model this through a Beta distribution modelling the uncertainty in the deployed $\hat{c}$ cost proportion conditional on the true c. This leads to a model with a single parameter γ, ranging from no uncertainty $\gamma = \infty$ to complete uncertainty $\gamma = 0$ on the operation condition. We have theoretically studied how uncertainty influences threshold choice methods. Concretely, we have introduced some results that show that with complete uncertainty, the threshold choice methods select cost lines proportionally to the importance (length of the the values of c where they are selected). These cost lines are used originally by these threshold choice methods (not considering uncertainty). We thus see that under high uncertainty, cost lines with optimal cost for a small cost range (cost lines with high slopes) could translate into high costs in some regions.

Hence, if our learning task presents uncertainty in the operating conditions, it might be better to use cost lines with lower slopes that do not imply a high cost for some ranges of the operating condition. We can see an example of this in the Support Vector Machine without scores of Fig. 7, where the score-driven threshold choice method obtains a better performance with complete uncertainty $\gamma = 0$ because it always selects a cost line with small slope in contrast to train-optimal, which, for the same example, uses three different cost lines. Experiments over 26 datasets and 12 learning methods show that, as expected, uncertainty degrades the performance of most learning techniques and threshold choice methods. Methods that estimate crisp probabilities (SVM not outputting scores, J48 unpruned or IBK1) are then recommended in situations with extreme uncertainty. This might be surprising but it conveys an important message that has been found whenever uncertainty appears in decision theory: perfectly rational procedures when unreliable information is assumed reliable may lead to suboptimal decisions. Only by including all the degrees of certainty involved can we expect rational decision making processes to be optimal.

There are many avenues of future work. For instance, we think that better versions of the rate-driven threshold choice method are possible. Alternatively, we think that with information of the uncertainty of the operating condition, we could define a new optimal threshold choice method for a given uncertainty, or even include the uncertainty as a new parameter in the score-driven method too. Finally, some of these ideas could be extended to multiclass classification, which would involve a more complex modelling of uncertainty.

In summary, this work closes a circle about the analysis of threshold choice methods and gives a realistic account of how operating conditions have to be used for classification, once one realises that they are rarely fully reliable. From this new understanding, we can even talk about a new kind of overfitting in machine learning, which depends on taking the operating condition (or a range of operating conditions) too seriously, as if it were infallible.

Acknowledgements We thank the anonymous reviewers for their comments, which have helped to improve this paper significantly. This work has been partially supported by the EU (FEDER) and the Spanish MINECO under Grant TIN 2015-69175-C4-1-R and by Generalitat Valenciana under Grant PROMETEOII/2015/013. José Hernández-Orallo was supported by a Salvador de Madariaga Grant (PRX17/00467) from the Spanish MECD for a research stay at the Leverhulme Centre for the Future of Intelligence (CFI), Cambridge, a BEST Grant (BEST/2017/045) from Generalitat Valenciana for another research stay also at the CFI and an FLI Grant RFP2-152.

Appendix A: Proofs

In this appendix, we give the proofs for the theorems, lemmas and corollaries in the paper.

Lemma 1 *Given $n + 1$ cost lines, the expected loss at point c is given by a weighted sum of the height of all $n + 1$ cost lines at c:*

$$Q(c) = \sum_{i=0}^{n} \Delta V_i(c) h_i(c)$$

where

$$\Delta V_i(c) = V(c_{i+1}|c) - V(c_i|c)$$

Proof For a cost proportion c and an uncertainty distribution v, the probability of the cost line i being selected by T is $\Delta V_i(c)$. Because of the linearity of cost lines the weights can be calculated as a difference between the cumulative distributions for the interval where T chooses i. □

Theorem 1 *The area under a cost curve (expected loss) can be decomposed as:*

$$L_v = \sum_{i=0}^{n} L_i$$

where

$$L_i = \int_0^1 [V(c_{i+1}|c) - V(c_i|c)]h_i(c)dc$$

quantifies the contribution of a single cost line to the expected loss.

Proof The proof is straightforward by regrouping the summands of lemma 1.

$$L_v = \int_0^1 Q(c)dc = \int_0^1 \sum_{i=0}^{n} \Delta V_i(c)h_i(c)dc = \sum_{i=0}^{n} \int_0^1 \Delta V_i(c)h_i(c)dc$$

□

Theorem 2 *The contribution of cost line i to the expected loss for complete uncertainty is* $L_i = (c_{i+1} - c_i)\{k_i/2 + l_i\}$.

Proof We can express the cost of that line as:

$$L_i = \int_0^1 [V(c_{i+1}|c) - V(c_i|c)](k_i c + l_i)dc \tag{12}$$

For complete uncertainty we have a uniform distribution, and the cumulative V is the identity function, so $V(c_{i+1}|c) = c_{i+1}$ and $V(c_i|c) = c_i$. Hence,

$$\begin{aligned} L_i &= \int_0^1 (c_{i+1} - c_i)(k_i c + l_i)dc \\ &= (c_{i+1} - c_i)\{k_i \left[c^2/2\right]_0^1 + [l_i c]_0^1\} \\ &= (c_{i+1} - c_i)\{k_i/2 + l_i\} \end{aligned}$$

□

Theorem 3 *The contribution of cost line i to the expected loss for complete certainty is* $L_i = (c_{i+1} - c_i)\{k_i(c_{i+1} + c_i)/2 + l_i\}$.

Proof For complete certainty the cumulative distribution is a step function at c, and the cost line has a non-zero contribution only if $c \in [c_i, c_{i+1}]$. Hence

$$\begin{aligned} L_i &= \int_{c_i}^{c_{i+1}} (k_i c + l_i)dc \\ &= \{k_i \left[c^2/2\right]_{c_i}^{c_{i+1}} + [l_i c]_{c_i}^{c_{i+1}}\} \\ &= k_i(c_{i+1}^2 - c_i^2)/2 + l_i(c_{i+1} - c_i) \\ &= (c_{i+1} - c_i)\{k_i(c_{i+1} + c_i)/2 + l_i\} \end{aligned}$$

□

Theorem 4 *Assuming scores in* $[0, 1]$, *the expected cost curve for the score-driven threshold choice method and complete operating condition uncertainty is given by:*

$$K^{sd}_{v[\gamma=0]}(c) = 2\{\pi_1(1-\overline{s}_1) + (\pi_0\overline{s}_0 - \pi_1(1-\overline{s}_1))c\} \tag{13}$$

where $\overline{s}_k$ *represents the average scores for class k.*

Proof We have $v_{[\gamma=0]}(\hat{c}|c) = 1$, so, from Eq. 8 and Definition 4 we derive:

$$\begin{aligned} K^{sd}_{v[\gamma=0]}(c) &= \int_0^1 Q(T^{sd}(\hat{c}); c) v_{[\gamma=0]}(\hat{c}|c) d\hat{c} \\ &= \int_0^1 Q(\hat{c}; c) d\hat{c} \\ &= \int_0^1 2\{c\pi_0(1-F_0(\hat{c})) + (1-c)\pi_1 F_1(\hat{c})\} d\hat{c} \\ &= \int_0^1 2\{\pi_1 F_1(\hat{c}) + (\pi_0(1-F_0(\hat{c})) - \pi_1 F_1(\hat{c}))c\} d\hat{c} \\ &= 2\{\pi_1(1-\overline{s}_1) + (\pi_0\overline{s}_0 - \pi_1(1-\overline{s}_1))c \end{aligned}$$

The last step follows because $\int_0^1 F_k(t)dt = [tF_k(t)]_0^1 - \int_0^1 tf_k(t)dt = 1 - \overline{s}_k$. □

Corollary 1 *The expected loss for the score-driven threshold choice method, uniform c and complete operating condition uncertainty is the classifier's mean absolute error:*

$$L^{sd}_{v[\gamma=0]} = \pi_0\overline{s}_0 + \pi_1(1-\overline{s}_1) = MAE.$$

Proof From Eq. 7,

$$\begin{aligned} L^{sd}_{v[\gamma=0]} &= \int_0^1 K^{sd}_{v[\gamma=0]}(c) w(c) dc \\ &= \int_0^1 2\{\pi_1(1-\overline{s}_1) + (\pi_0\overline{s}_0 - \pi_1(1-\overline{s}_1))c\} w(c) dc \\ &= 2\{\pi_1(1-\overline{s}_1) + (\pi_0\overline{s}_0 - \pi_1(1-\overline{s}_1))\mathbb{E}_w\{c\}\} \end{aligned}$$

For uniform c we have $\mathbb{E}_w\{c\} = 1/2$ and the result follows. □

Theorem 5 *The expected cost curve for the rate-driven threshold choice method and complete operating condition uncertainty is given by:*

$$K^{rd}_{v[\gamma=0]}(c) = 2\pi_0\pi_1(1-AUC) + \pi_1^2 + (\pi_0 - \pi_1)c \tag{14}$$

Proof We have $v_{[\gamma=0]}(\hat{c}|c) = 1$, so, from Eq. 8 and Definition 5 we derive:

$$K^{rd}_{v[\gamma=0]}(c) = \int_0^1 Q(T^{rd}(\hat{c}); c) v_{[\gamma=0]}(\hat{c}|c) d\hat{c}$$

$$
\begin{aligned}
&= \int_0^1 Q(R^{-1}(\hat{c}); c) d\hat{c} \\
&= 2 \int_0^1 \{c\pi_0(1 - F_0(R^{-1}(\hat{c}))) + (1 - c)\pi_1 F_1(R^{-1}(\hat{c}))\} d\hat{c} \\
&= 2 \int_0^1 \{c\pi_0 - c\pi_0 F_0(R^{-1}(\hat{c})) + \pi_1 F_1(R^{-1}(\hat{c})) - c\pi_1 F_1(R^{-1}(\hat{c}))\} d\hat{c}.
\end{aligned}
$$

Since, by the definition of R given in the preliminaries, $\pi_0 F_0(R^{-1}(\hat{c})) + \pi_1 F_1(R^{-1}(\hat{c})) = R(R^{-1}(\hat{c})) = \hat{c}$,

$$
\begin{aligned}
K^{rd}_{v[\gamma=0]}(c) &= 2 \int_0^1 \{c\pi_0 - c\hat{c} + \pi_1 F_1(R^{-1}(\hat{c}))\} d\hat{c} \\
&= 2c\pi_0 - c + 2\pi_1 \int_0^1 \{F_1(R^{-1}(\hat{c}))\} d\hat{c}
\end{aligned}
$$

From the proof of (Hernández-Orallo et al. 2013, Theorem 22) we have

$$
2\pi_1 \int_0^1 F_1(R^{-1}(c)) dc = 2\pi_0\pi_1(1 - AUC) + \pi_1^2
$$

and so

$$
\begin{aligned}
K^{rd}_{v}(c) &= 2\pi_0\pi_1(1 - AUC) + \pi_1^2 + (2\pi_0 - 1)c \\
&= 2\pi_0\pi_1(1 - AUC) + \pi_1^2 + (\pi_0 - \pi_1)c
\end{aligned}
$$

□

Corollary 2 *The expected loss for the rate-driven threshold choice method, uniform c and complete operating condition uncertainty is related to the classifier's AUC as follows:*

$$
L^{rd}_{v[\gamma=0]} = \pi_0\pi_1(1 - 2AUC) + 1/2
$$

which is 1/6 *higher than if the operating condition were fully known.*

Proof

$$
\begin{aligned}
L^{rd}_{v[\gamma=0]} &= \int_0^1 K^{rd}_{v[\gamma=0]}(c) w(c) dc \\
&= \int_0^1 \{2\pi_0\pi_1(1 - AUC) + \pi_1^2 + (\pi_0 - \pi_1)c\} w(c) dc \\
&= 2\pi_0\pi_1(1 - AUC) + \pi_1^2 + (\pi_0 - \pi_1)\mathbb{E}_w\{c\}
\end{aligned}
$$

For uniform c we have $\mathbb{E}_w\{c\} = 1/2$ and so

$$\begin{aligned} L^{rd}_{v[\gamma=0]} &= 2\pi_0\pi_1(1 - AUC) + \pi_1^2 + (\pi_0 - \pi_1)/2 \\ &= \pi_0\pi_1(1 - 2AUC) + \pi_0\pi_1 + (1 - \pi_0)\pi_1 + (\pi_0 - \pi_1)/2 \\ &= \pi_0\pi_1(1 - 2AUC) + \pi_1 + (\pi_0 - \pi_1)/2 \\ &= \pi_0\pi_1(1 - 2AUC) + (\pi_0 + \pi_1)/2 \\ &= \pi_0\pi_1(1 - 2AUC) + 1/2 \end{aligned}$$

(Hernández-Orallo et al. 2013, Theorem 22) derive the expected loss over uniform c without considering operating condition uncertainty as $L^{rd}_{U(c)} = \pi_0\pi_1(1 - 2AUC) + 1/3$. □

Theorem 6 *The expected cost curve for the optimal threshold choice method and complete operating condition uncertainty is:*

$$K^o_{v[\gamma=0]}(c) = 2\pi_0\overline{s}^*_0$$

where s^* denotes scores obtained after perfect calibration. Hence the loss is independent of the true operating condition c.

Proof From Eq. 8 and Definition 3 we derive:

$$K^o_{v[\gamma=0]}(c) = \int_0^1 Q(T^o(\hat{c}); c)v_{[\gamma=0]}(\hat{c}|c)d\hat{c}$$

We know from Theorem 44 of Hernández-Orallo et al. (2013) that T^o is equivalent to T^{sd} for calibrated models. In other words, optimal thresholds are equivalent to score-driven thresholds after perfect calibration. Hence from Theorem 4 we derive:

$$K^o_{v[\gamma=0]}(c) = 2c\pi_0\overline{s}^*_0 + 2(1 - c)\pi_1(1 - \overline{s}^*_1)$$

Furthermore, Theorem 41 of Hernández-Orallo et al. (2013) states that for calibrated scores $\pi_0\overline{s}_0 = \pi_1(1 - \overline{s}_1)$, and the result follows. □

Theorem 7 *The expected loss for the optimal threshold choice method with complete uncertainty is twice the expected loss for the optimal threshold choice method with complete certainty:*

$$L^o_{v[\gamma=0]} = 2L^o_\delta \tag{15}$$

Proof We make the model m explicit as an argument and we will denote the model resulting from calibration as $\mathrm{Cal}(m)$. After perfect calibration, we know that optimal thresholds are equivalent to score-driven thresholds (according to Theorem 44 of Hernández-Orallo et al. (2013)), so we have:

$$L^o_{v[\gamma=0]}(m) = L^{sd}_{v[\gamma=0]}(\mathrm{Cal}(m))$$

From Theorem 4, we have:

$$L^{sd}_{v[\gamma=0]}(\mathrm{Cal}(m)) = L^{su}_\delta(\mathrm{Cal}(m)$$

Now, according to Figure 4 in Hernández-Orallo et al. (2013) we can follow all these equalities for calibrated classifiers:

$$L^{su}_\delta(\mathrm{Cal}(m)) = MAE(\mathrm{Cal}(m)) = 2RL(\mathrm{Cal}(m))$$

Here, RL refers to the refinement loss of a classifier, which is one of the components of the Brier score of a classifier (the other component being the calibration loss, which is zero for a perfectly calibrated classifier).

Similarly for the right-hand side of the theorem:

$$L^o_\delta(m) = L^{sd}_\delta(\mathrm{Cal}(m))$$

and again referring to Figure 4 in Hernández-Orallo et al. (2013):

$$L^{sd}_\delta(\mathrm{Cal}(m)) = BS(\mathrm{Cal}(m)) = RL(\mathrm{Cal}(m)$$

□

Appendix B: Threshold choice methods on a toy example

We give an example of using different threshold choice methods with complete certainty and complete uncertainty.

We trained a decision tree model (J48 Unpruned) for the Spambase dataset of the UCI repository Lichman (2013). This dataset contains a collection of spam and valid emails, and it has been used to build spam filters. This dataset contains 4554 instances. The model was trained with 1% of the data (47 instances, 28 of class 0 and 19 of class 1). The following text shows the induced decision tree.

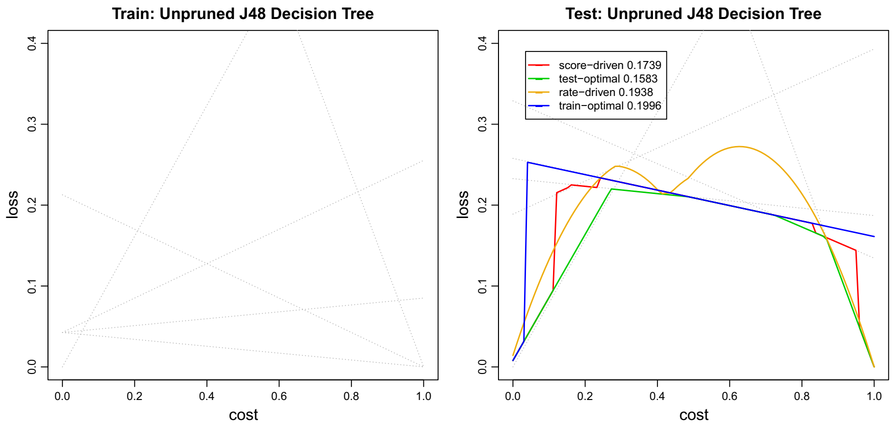


Fig. 8 Left: Cost lines of the J48 unpruned model in the train dataset. Right: Cost lines of the J48 unpruned model in the test dataset. Cost curves by four different threshold choice methods: score-driven, test-optimal, rate-driven and train-optimal. We consider the actual classes of the left example for the train-optimal threshold choice method. The area under the four cost curves is shown in the legend

```
word_freq_our <= 0.28
|   word_freq_your <= 3.17
|   |   char_freq_! <= 0.204: 0 (23/0)
|   |   char_freq_! > 0.204
|   |   |   capital_run_length_longest <= 10: 0 (4/0)
|   |   |   capital_run_length_longest > 10: 1 (0/4)
|   word_freq_your > 3.17: 1 (0/2)
word_freq_our > 0.28: 1 (1/13)
```

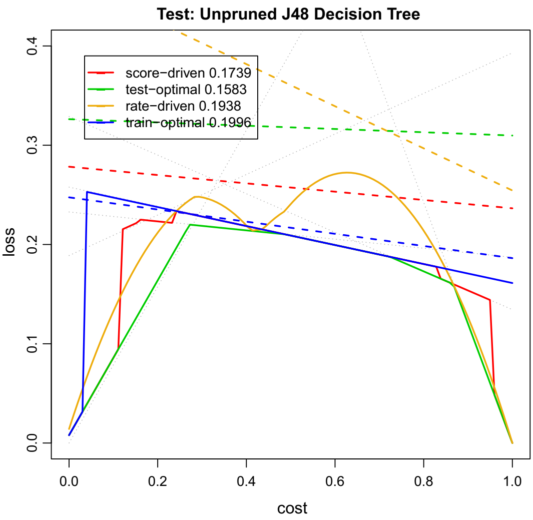
There are five leaves in the tree, the distribution of positive and negative examples is included in the model (numbers between parenthesis). Since Laplace correction is applied, the estimated scores of being class 1 for each of the leaves are: (0.040, 0.833, 0.167, 0.750, 0.875). Higher scores express a stronger belief that the instance is of class 1.

We have 5 leaves and hence 6 cost lines $Q(t; 0) \rightarrow Q(t; 1)$, as represented by Fig. 8 (left), namely $(i = 0)$ $0 \rightarrow 0.8085$, $(i = 1)$ $0.0425 \rightarrow 0.2553$, $(i = 2)$ $0.0425 \rightarrow 0.0851$, $(i = 3)$ $0.0425 \rightarrow 0$, $(i = 4)$ $0.2127 \rightarrow 0$ and $(i = 5)$ $1.1914 \rightarrow 0$. If we set a fixed threshold on $t = 0.5$, we would have a cost curve that corresponds to the cost line in the plot ($i = 3$, $0.0425 \rightarrow 0$).

More generally, if we change the threshold with c we would obtain a cost curve that jumps between these cost lines in a way determined by the threshold choice method. In particular, the lowest two cost lines cross at around $c = 0.05$ and so the optimal strategy would be to choose the $(i = 0)$ cost line for $c < 0.05$ and the $(i = 3)$ cost line for $c > 0.05$. In terms of thresholds, this would be achieved by setting t such that $t < 0.04$ for $c < 0.05$ and $0.167 < t < 0.75$ for $c > 0.05$.

Now consider applying the same model to the test dataset with 4554 examples (2760 of class 0 and 1794 of class 1). The previous model obtains an Area under the ROC curve of 0.7921, a Mean Average Error (MAE) of 0.2546, and a Mean Square

Fig. 9 J48 unpruned decision tree for the Spambase dataset with cost curves under complete operating condition certainty (solid lines) and complete uncertainty (dashed lines)



Error (MSE) of 0.1743. On this test set, the 6 cost lines are $(i = 0)$ 0 $\rightarrow$ 0.788, $(i = 1)$ 0.189 $\rightarrow$ 0.393, $(i = 2)$ 0.232 $\rightarrow$ 0.187, $(i = 3)$ 0.258 $\rightarrow$ 0.161, $(i = 4)$ 0.329 $\rightarrow$ 0.134, and $(i = 5)$ 1.21 $\rightarrow$ 0. These cost lines can be seen in the right plot of Fig. 8. On this test set, the optimal strategy would be select cost line $i = 0$ for $c < 0.27$, cost line $i = 2$ for $0.27 < c < 0.5$, cost line $i = 3$ for $0.5 < c < 0.72$, cost line $i = 4$ for $0.72 < c < 0.84$, and cost line $i = 5$ for $c > 0.84$ with an expected loss 0.1583; but this *test-optimal* strategy would require knowledge of the test labels. If we apply the *train-optimal* strategy from before we see that suboptimal decisions are made in the range $0.05 < c < 0.27$ and $c > 0.84$, resulting in a higher expected loss of 0.1977. This difference between train-optimal and test-optimal has usually been overlooked in the literature.

Threshold choice methods on a toy example with uncertainty

Table 3 summarises our results so far for complete operating condition uncertainty. The first column gives the expected cost lines in the form $K(c) = A + Bc$, and the right column gives the expected loss (area under the expected cost line). Table 4 gives the corresponding results for complete operating condition certainty from Hernández-Orallo et al. (2012).

We start by considering the example of Fig. 8. Figure 9 compares the cost curves for complete operating condition certainty (solid lines) and complete uncertainty (dashed lines).

Score-driven: In the test set we have $\pi_0 = 0.394$, $\pi_1 = 0.606$, $\overline{s}_0 = 0.3065$ and $\overline{s}_1 = 0.7790$, so, following the equations in Table 3, the score-driven cost curve under complete uncertainty can be derived as $K(c) = 0.2677 - 0.0263c$. The area under this curve is $MAE = 0.2546$, which is worse that of the curve under complete certainty ($BS = 0.1743$). Yet, as can be seen in Fig. 9 there is an interval around

$c = 0.2$ where the model is poorly calibrated and the expected loss under complete uncertainty is lower than the expected loss under complete certainty.

Rate-driven: We have $AUC = 0.7921$ and so the equation of the rate-driven cost curve under complete uncertainty can be derived as $K(c) = 0.4665 - 0.21c$ with an area of 0.3605, which is $1/6$ higher than the expected loss under complete certainty (0.1938).

Test-optimal: We can follow the calculations for score-driven thresholds if we first calibrate the model, obtaining calibrated scores s^*. Here the optimal cost curve under complete uncertainty has an expected loss of 0.3166, which is twice the expected loss under complete certainty (0.1583).

Train-optimal: Again, we need to calibrate scores, and then we can follow the calculations for score-driven. The train-optimal cost curve under complete uncertainty obtains an expected loss of 0.2168, which should be contrasted with the expected loss under complete certainty (0.1977).

In general, we see that, as expected, uncertainty degrades the expected loss for all the threshold choice methods. However, if we observe the particular curve train-optimal, we can find a small area where the complete uncertainty curve dips below the curves for complete certainty. Hence, it may happen that a bad choice of threshold is mitigated by a bad estimation of the operating condition.

References

Adams N, Hand D (1999) Comparing classifiers when the misallocation costs are uncertain. Pattern Recognit 32(7):1139–1147

Bella A, Ferri C, Hernández-Orallo J, Ramírez-Quintana MJ (2013) On the effect of calibration in classifier combination. Appl Intell 38(4):566–585

Bishop C (2011) Embracing uncertainty: applied machine learning comes of age. In: Machine learning and knowledge discovery in databases. Springer, Berlin, pp 4

Brier GW (1950) Verification of forecasts expressed in terms of probability. Monthly Weather Rev 78(1):1–3

Dalton LA (2016) Optimal ROC-based classification and performance analysis under Bayesian uncertainty models. IEEE/ACM Trans Comput Biol Bioinform (TCBB) 13(4):719–729

de Melo C, Eduardo C, Bastos Cavalcante Prudencio R (2014) Cost-sensitive measures of algorithm similarity for meta-learning. In: 2014 Brazilian conference on intelligent systems (BRACIS). IEEE, pp 7–12

Dou H, Yang X, Song X, Yu H, Wu WZ, Yang J (2016) Decision-theoretic rough set: a multicost strategy. Knowl-Based Syst 91:71–83

Drummond C, Holte RC (2000) Explicitly representing expected cost: an alternative to roc representation. In: Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, NY, USA, KDD '00, pp 198–207

Drummond C, Holte RC (2006) Cost curves: an improved method for visualizing classifier performance. Mach Learn 65(1):95–130

Elkan C (2001) The foundations of cost-sensitive learning. In: Proceedings of the 17th international joint conference on artificial intelligence, vol 2. Morgan Kaufmann Publishers Inc., IJCAI'01, pp 973–978

Fawcett T (2003) In vivo spam filtering: a challenge problem for KDD. ACM SIGKDD Explor. Newsl. 5(2):140–148

Fawcett T (2006) An introduction to ROC analysis. Pattern Recognit Lett 27(8):861–874

Fawcett T, Niculescu-Mizil A (2007) PAV and the ROC convex hull. Mach Learn 68(1):97–106

Ferri C, Flach PA, Hernández-Orallo J (2017) R code for threshold choice methods with context uncertainty. https://github.com/ceferra/ThresholdChoiceMethods/tree/master/Uncertainty

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

Flach P (2014) Classification in context: adapting to changes in class and cost distribution. In: First international workshop on learning over multiple contexts at European conference on machine learning and principles and practice of knowledge discovery in databases ECML-PKDD'2014

Flach P, Matsubara ET (2007) A simple lexicographic ranker and probability estimator. In: 18th European conference on machine learning, ECML2007. Springer, pp 575–582

Flach P, Hernández-Orallo J, Ferri C (2011) A coherent interpretation of AUC as a measure of aggregated classification performance. In: Proceedings of the 28th international conference on machine learning, ICML2011

Guzella TS, Caminhas WM (2009) A review of machine learning approaches to spam filtering. Expert Syst Appl 36(7):10206–10222

Hand D (2009) Measuring classifier performance: a coherent alternative to the area under the ROC curve. Mach Learn 77(1):103–123

Hernández-Orallo J, Flach P, Ferri C (2011) Brier curves: a new cost-based visualisation of classifier performance. In: Proceedings of the 28th international conference on machine learning, ICML2011

Hernández-Orallo J, Flach P, Ferri C (2012) A unified view of performance metrics: translating threshold choice into expected classification loss. J Mach Learn Res 13(1):2813–2869

Hernández-Orallo J, Flach P, Ferri C (2013) ROC curves in cost space. Mach Learn 93(1):71–91

Hornik K, Buchta C, Zeileis A (2009) Open-source machine learning: R meets Weka. Comput Stat 24(2):225–232

Huang Y (2015) Dynamic cost-sensitive naive bayes classification for uncertain data. Int J Database Theory Appl 8(1):271–280

Johnson RA, Raeder T, Chawla NV (2015) Optimizing classifiers for hypothetical scenarios. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, Berlin, pp 264–276

Lichman M (2013) UCI machine learning repository. http://archive.ics.uci.edu/ml

Liu M, Zhang Y, Zhang X, Wang Y (2011) Cost-sensitive decision tree for uncertain data. In: Advanced data mining and applications. Springer, Berlin, pp 243–255

Liu XY, Zhou ZH (2010) Learning with cost intervals. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 403–412

Provost F, Fawcett T (2001) Robust classification for imprecise environments. Mach Learn 42(3):203–231

Provost FJ, Fawcett T et al (1997) Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. KDD 97:43–48

Qin B, Xia Y, Li F (2009) DTU: a decision tree for uncertain data. In: Advances in knowledge discovery and data mining. Springer, Berlin, pp 4–15

Ren J, Lee SD, Chen X, Kao B, Cheng R, Cheung D (2009) Naive Bayes classification of uncertain data. In: Ninth IEEE international conference on data mining, 2009. ICDM'09. IEEE, pp 944–949

Ridzuan F, Potdar V, Talevski A (2010) Factors involved in estimating cost of email spam. In: Taniar D, Gervasi O, Murgante B, Pardede E, Apduhan BO (eds) Computational science and its applications—ICCSA 2010. Springer, Berlin, pp 383–399

Sakkis G, Androutsopoulos I, Paliouras G, Karkaletsis V, Spyropoulos CD, Stamatopoulos P (2003) A memory-based approach to anti-spam filtering for mailing lists. Inf Retr 6(1):49–73

Tsang S, Kao B, Yip KY, Ho WS, Lee SD (2011) Decision trees for uncertain data. IEEE Trans Knowl Data Eng 23(1):64–78

Wang R, Tang K (2012) Minimax classifier for uncertain costs. arXiv preprint arXiv:1205.0406

Zadrozny B, Elkan C (2001a) Learning and making decisions when costs and probabilities are both unknown. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 204–213

Zadrozny B, Elkan C (2001b) Obtaining calibrated probability estimates from decision trees and Naive Bayesian classifiers. In: Proceedings of the eighteenth international conference on machine learning (ICML 2001), pp 609–616

Affiliations

Cèsar Ferri[1] · José Hernández-Orallo[1] · Peter Flach[2]

Cèsar Ferri
cferri@dsic.upv.es

José Hernández-Orallo
jorallo@dsic.upv.es

[1] Universitat Politècnica de València, Camí de Vera s/n, 46022 Valencia, Spain

[2] Intelligent Systems Laboratory, University of Bristol, Merchant Venturers Building, Woodland Road, Bristol BS8 1UB, UK