Cost Sensitive Evaluation of Instance Hardness in Machine Learning

Ricardo B. C. Prudêncio (✉)

Centro de Informática, Universidade Federal de Pernambuco, Brazil
rbcp@cin.ufpe.br

Abstract. Measuring hardness of individual instances in machine learning contributes to a deeper analysis of learning performance. This work proposes instance hardness measures for binary classification in cost-sensitive scenarios. Here cost curves are generated for each instance, defined as the loss observed for a pool of learning models for that instance along the range of cost proportions. Instance hardness is defined as the area under the cost curves and can be seen as an expected loss of difficulty along cost proportions. Different cost curves were proposed by considering common decision threshold choice methods in literature, thus providing alternative views of instance hardness.

1 Introduction

Measuring difficulty in machine learning (ML) strongly contributes to understanding the potential advantages and limitations of the learning algorithms. Previous work has mainly focused on deriving complexity measures for datasets [1, 7, 14]. Alternatively, the current work follows the instance-level approach, focused on measuring hardness for individual instances. Instance hardness measures can be useful to a deeper analysis of algorithm performance and to investigate specific causes of bad learning behavior [17, 12]. Distinct areas of ML have developed methods which somehow rely on measuring difficulty of instances (e.g., dynamic classifier selection [20, 19, 4], noise detection [3, 18, 16] and active learning [13]).

In [17, 16, 11], instance hardness is defined based on the learning behavior of a pool of algorithms (e.g., the proportion of algorithms that misclassified the instance). In [15], the authors addressed instance difficulty by proposing four types of examples: safe (easy instances), borderline, rare and outliers (difficult instances). Each instance is categorized into a difficulty type by considering the distribution of classes in the neighborhood of the instance. However, these straightforward ideas do not consider an important practical issue, which is the cost associated to the classifier errors [5]. The costs of false positives and false negatives may vary at deployment time. In this sense, misclassification in specific areas of the instance space may have more significance. Instance hardness measures should identify such areas by defining difficulty not only in terms of observed errors, but also in terms of expected costs.

Additionally, in cost-sensitive scenarios, when a model returns scores (e.g., class probabilities), decision thresholds can be adapted according to the error costs. For instance, when the cost of false negatives is higher than false positives, the threshold can be set to increase the number of positive predictions. In [9], the loss of a model depends on the *threshold choice method* (TCM) adopted. Yet, model performance for instances may vary too, requiring new hardness measures.

This work proposes a new framework to measure instance hardness for binary classification problems in cost-sensitive scenarios. Initially, the concept of *instance cost curve* is proposed, which plots the loss produced by a model for that instance along the cost proportions. A different instance cost curve is produced for each different TCM. This is a new concept which extends previous work on cost curves, now aiming to evaluate and inspect loss for individual instances. Instance cost curves were derived for five different TCMs: score-fixed, score-driven, rate-driven, score-uniform and rate-uniform methods [9].

By plotting an instance cost curve, one can visualize how difficult the instance is for each cost proportion. A global instance hardness measure can be defined as the area under the cost curve (i.e., the expected loss obtained for a learned model for an instance along the range of cost proportions). In order to avoid defining instance hardness based upon a single model, the ensemble strategy proposed in [17] was adopted here. More specifically, a set of instance cost curves is generated using a pool of learned models and the average instance hardness is computed.

The proposed framework addresses different issues. First, it is possible to identify the hard instances in a problem and under which operation conditions (cost proportions) they are difficult. The use of different TCMs provides new perspectives for measuring hardness, including misclassification evaluation, probability estimation and ranking performance. Yet, for some TCMs, hardness can be measured under cost proportion uncertainty. The instance-level approach also supports the development of hardness measures for groups of instances and particularly class hardness measures. Different ML areas which already use instance hardness measures can benefit from the proposed framework. The adequate hardness measure must be chosen depending on the application objectives. For instance, if one wants to improve class probability estimation, a hardness measure based on scores should be adopted. We believe that such areas can be extended more adequately to cost-sensitive scenarios by adopting the proposed measures.

2 Notation and Basic Definitions

The basic notation adopted in this work is based on [9]. Instances are classified into one of the classes $Y = \{0, 1\}$, in which 0 is the positive class and 1 is the negative class. A model m is a scoring function that receives an instance x as input and returns a score $s = m(x)$ indicating the chance of a negative class prediction. A model is transformed into a classifier assuming a decision threshold t. If $s \leq t$ then x is classified as positive and classified as negative otherwise.

The classifier errors can be associated to different costs. The cost of a *false negative* is c_0, while the cost of a *false positive* is c_1. As in [9], the costs are normalized by setting $c_0 + c_1 = b$ and the *cost proportion* $c = c_0/b$ represents the operating condition faced by a model when it is deployed. For simplicity, this work adopted $b = 2$ and hence $c \in [0, 1]$, $c_0 = 2c$ and $c_1 = 2(1 - c)$.

Let $f_0(s)$ and $f_1(s)$ be the score density functions respectively for the positive and negative classes. The *false negative* rate obtained by setting a threshold t is defined as $F_0(t) = \int_{-\infty}^{t} f_0(s)ds$. The *false positive* rate, in turn, is defined as $F_1(t) = \int_{-\infty}^{t} f_1(s)ds$. The positive rate $R(t)$ (i.e., the proportion of instances predicted as positive) is $R(t) = \pi_0(1 - F_0(t)) + \pi_1 F_1(t)$, in which π_0 and π_1 are the proportions of positive and negative examples. The loss for a threshold t and a cost proportion c is defined as:

$$\begin{aligned} Q(t, c) &= c_0 \pi_0 F_0(t) + c_1 \pi_1 F_1(t) \\ &= 2\{c\pi_0 F_1(t) + (1 - c)\pi_1 F_1(t)\} \end{aligned} \tag{1}$$

A *threshold choice method* (TCM) is a function $T(c)$ which defines the decision threshold according to the operation condition. The expected loss of a model can be expressed as Eq. 2, in which $w_c(c)$ is the distribution of cost proportions:

$$L = \int_0^1 Q(T(c), c) w_c(c) dc \tag{2}$$

3 Instance Hardness and Cost Curves

By assuming uniform distribution of operation conditions, in [9] it is proved that the loss L is directly related to different performance measures depending on the TCM. If the threshold is fixed (e.g., 0.5) regardless c, L is the error rate at that threshold. Under the score-driven TCM (i.e., T(c) = c), in turn, the loss is equal to the Brier score of the model. Under the rate-driven method, when a threshold is set to obtain a desired positive prediction rate, the loss is linearly related to AUC. The appropriate measure depends on the cost-sensitive scenario.

Similarly, instance hardness may depend on the TCM. For instance, consider three positive instances with scores 0.2, 0.6 and 0.8. The 1st instance is correctly classified if a fixed $t = 0.5$ is adopted, while the 2nd and 3rd instances are false negatives. In this case, instance hardness depends solely on the threshold and the score. In case $T(c) = c$ is adopted, the 1st instance is very easy since it is correctly classified in a wide range of operation conditions. Yet, the 3rd instance is harder than the 2nd one. Here, hardness also depends on the operation condition.

This paper proposes a new framework for instance hardness evaluation which takes the above nuances into account. The expected model loss expressed in Eq. 2 is an aggregation over the operation conditions. The main idea is to transform the loss function to be expressed as an aggregation over scores (instead of costs) and then to define the contribution of each instance in the model loss. Initially, $Q(t, c)$ (Eq. 1) is decomposed into two functions respectively for false negatives and false positives. For false negatives: $Q_0(t, c) = 2c\pi_0(1 - F_0(t))$. After some algebraic operations, this term is defined as an integral over scores:

$$\begin{aligned} Q_0(t,c) &= 2c\pi_0(1 - F_0(t)) \\ &= 2c\pi_0(1 - \textstyle\int_{-\infty}^{t} f_0(s)ds) \\ &= 2c\pi_0(\textstyle\int_s f_0(s)ds - \int_s \delta(s,t)f_0(s)ds) \\ &= 2c\pi_0(\textstyle\int_s (1 - \delta(s,t))f_0(s)ds \\ &= \textstyle\int_s 2c\pi_0(1 - \delta(s,t))f_0(s)ds \end{aligned} \tag{3}$$

where $\delta(s,t) = 1$ if $s \leq t$ and $= 0$ otherwise. Notice that a false negative occurs when the instance is positive and $1 - \delta(s,t) = 1$, i.e., $s > t$. The expected loss of the positive class over the operation conditions can be expressed as:

$$\begin{aligned} L_0 &= \textstyle\int_c Q_0(t,c)dc \\ &= \textstyle\int_c \int_s 2c\pi_0(1 - \delta(s,t))f_0(s)dsdc \\ &= \textstyle\int_s \int_c \pi_0 f_0(s)\mathbf{2c}(\mathbf{1} - \boldsymbol{\delta}(\mathbf{s},\mathbf{t}))dcds \end{aligned} \tag{4}$$

In Eq. 4, a positive instance is associated to a loss $2c$ when it is incorrectly classified, i.e., when $1-\delta(s,t) = 1$. Otherwise, the loss is zero. Then, the *instance cost curve* for a positive instance with score s is defined as:

$$QI_0(s,t,c) = 2c(1 - \delta(s,t)) \tag{5}$$

Depending on the TCM, different curves can be produced along c. *Instance hardness* is then defined as the area under the instance cost curve (the expected loss for the range of operation conditions). In general, given a TCM $T(c)$, the hardness of a positive instance with score s is:

$$IH_0^T(s) = \int_c QI_0(s, T(c), c)dc \tag{6}$$

By replacing the instance hardness Eq. 6 in Eq. 4, the expected loss for the positive class is alternatively defined as an aggregation of hardness over the distribution of scores:

$$L_0 = \pi_0 \int_s IH_0^T(s) f_0(s)ds \tag{7}$$

A similar derivation can be performed in order to define instance cost curves and hardness values for negative instances. An error for a negative instance occurs when $\delta(s,t) = 1$ and the associated loss is $2(1 - c)$. The instance cost curve for a negative instance with score s is defined as:

$$QI_1(s,t,c) = 2(1 - c)\delta(s,t) \tag{8}$$

Instance hardness assuming a function $T(c)$ and the loss relative to the negative class is defined as:

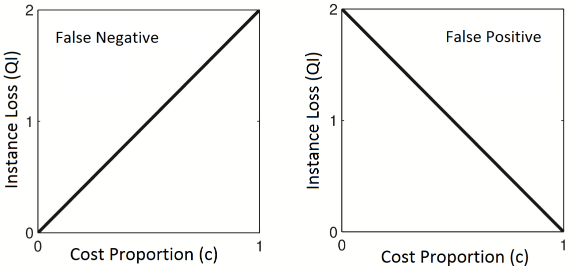

Fig. 1. Instance cost curves assuming the SF method.

$$IH_1^T(s) = \int_c QI_1(s, T(c), c)dc \tag{9}$$

$$L_1 = \pi_1 \int_s IH_1^T(s) f_1(s) ds \tag{10}$$

In this work, the hardness measures for five TCMs [10] were derived. For robustness, as in [17], a set of models can be used to compute the average hardness across models. All implementations are provided in an online material[1].

3.1 Score-Fixed Instance Hardness

The *score-fixed* (SF) method assumes a fixed threshold regardless the condition c. Typically, t is set to 0.5. Consider a positive instance with score $s > t$. This instance is always a false negative regardless c, as the threshold is fixed. In this case, $\delta(s,t) = 0$. By replacing it in Eq. 5, the instance cost curve is defined as:

$$QI_0(s, t, c) = 2c \tag{11}$$

In turn, the cost curve for a false positive instance is:

$$QI_1(s, t, c) = 2(1 - c) \tag{12}$$

Fig. 1 illustrates the SF instance cost curves for false negatives and false positives. For correctly classified instances, the cost curve is just a constant line $QI(s, t, c) = 0$. By integrating QI, the instance hardness values respectively for false negatives and false positives are derived as follows:

$$IH_0^{sf}(x) = \int_0^1 2c\,\mathrm{d}c = \Big[c^2\Big]_0^1 = 1 \tag{13}$$

$$IH_1^{sf}(x) = \int_0^1 2(1-c)\,\mathrm{d}c = \Big[2c - c^2\Big]_0^1 = 1 \tag{14}$$

[1] https://tinyurl.com/y3cthlv8

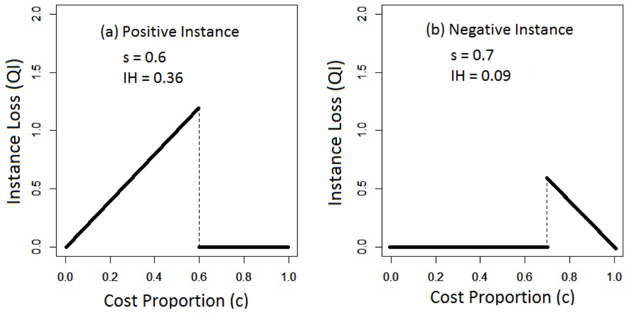

Fig. 2. Instance cost curves assuming the SD method.

For correctly classified instances (either positive or negative), $IH^{sf}(x) = 0$. The SF hardness is simply the $0|1$ loss. By adopting a pool of models, instance hardness is the proportion of incorrect classifications provided by the pool.

3.2 Score-Driven Instance Hardness

Although SF is frequently used, when the classifier errors have different costs, it is sound to assign thresholds accordingly [6]. In the *score-driven* (SD) TCM [8], the threshold is set to c (i.e., $T(c) = c$). For instance, if $c = 0.7$, the cost of false negatives is high. By setting $t = 0.7$, the classifier predicts more instances as positive, minimizing the number of false negatives. In the SD method, a positive instance is predicted as negative when $s > c$ and correctly predicted otherwise. Then $\delta(s,t) = 0$ if $s > c$, which results in the following instance cost curve (Eq. 15) by replacing $\delta(s,t)$ in Eq. 5. The area under the curve is defined in Eq. 16. Fig. 2(a) illustrates the SD cost curve for a positive instance with $s = 0.6$.

$$QI_0(s,t,c) = \begin{cases} 2c, & \text{if } s > c \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

$$IH^{sd}(x) = \int_0^s 2c\,\mathrm{d}c = \Big[c^2\Big]_0^s = s^2 \tag{16}$$

Since $y = 0$ for positive instances, the above measure can be replaced by $(y - s)^2$, which is the squared-error of the model. For negative instances, Eq. 17 and 18 define the cost curve and hardness measure. Fig. 2(b) illustrates the curve for a negative instance with $s = 0.7$. For negative instances, $y = 1$. Again hardness corresponds to $(y - s)^2$, the squared-error. When the ensemble is adopted, the hardness of an instance is the average squared-error obtained by the pool.

$$QI_1(s,t,c) = \begin{cases} 2(1-c), & \text{if } s \leq c \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

$$IH^{sd}(x) = \int_s^1 2(1-c)\,\mathrm{d}c = \Big[2c - c^2\Big]_s^1 = (1-s)^2 \tag{18}$$

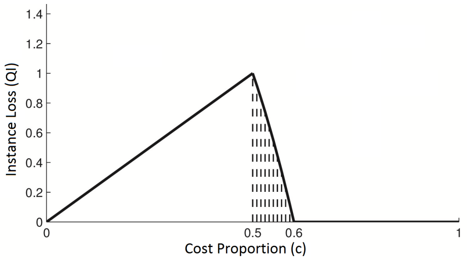

Fig. 3. Instance cost curve for a positive instance - RD method.

3.3 Rate-Driven Instance Hardness

The SD method is a natural choice when the model is assumed to be a class probability estimator. However, SD is sensitive to the score estimation [9]. If scores are highly concentrated, a small change in operating condition (and in the threshold) may drastically affect performance. As an alternative, the positive rate $R(t)$ can be used to define thresholds [10]. In the *rate-driven* (RD) method, the threshold is set to achieve a desired positive rate, i.e., $T^{rd}(c) = R^{-1}(c)$. For instance, if $c = 0.7$ the threshold t is set in such a way that 70% of the instances are classified as positive. The operating condition c is then expressed as the desired positive rate: $c = R(t)$. Scores can be seen as rank indicators instead of probabilities. The RD cost curve for a positive instance is defined as:

$$QI_0(s,t,c) = \begin{cases} 2c, & \text{if } s > R^{-1}(c) \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

For $R(s) \leq c$ (equivalent to $s \leq R^{-1}(c)$) loss is zero. When $R(s) > c$, the loss varies linearly. The RD hardness is defined in Eq. 20, which is related to the position of the instance in the ranking produced by the model (i.e., $R(s)$). Different from SD, which measures error, RD measures ranking performance. A hard instance for SD may be easy for RD depending on the score distribution.

$$IH^{rd}(x) = \int_0^{R(s)} 2c\,\mathrm{d}c = \left[c^2\right]_0^{R(s)} = R(s)^2 \tag{20}$$

An adjustment is necessary when the cost curve is built for real datasets. In such case, the range of desired positive rates is continuous, whereas the number of observed rates is limited by the dataset size. Fig. 3 shows the cost curve for x_6 and model m_1 in Table 1. The positive rate of x_6 is 0.6, i.e., $R(0.75) = 0.6$. The previous observed positive rate is 0.5 assuming the previous score 0.7 as threshold ($R(0.7) = 0.5$). Instance x_6 is correctly classified if the desired positive rate is equal or higher than 0.6, (loss is zero for $c \in [0.6; 1]$). For $c < 0.5$, the instance is classified as negative and its loss varies linearly. Positive rates between 0.5 and 0.6 can not be produced using m_1. In such cases, the loss is estimated from stochastic interpolation between 0.5 and 0.6 (dashed area in Fig. 3).

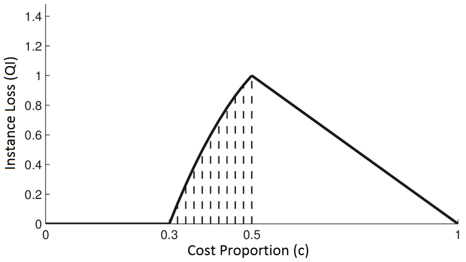

Fig. 4. Instance cost curve for a negative instance - RD method.

Table 1. Example of instances and scores provided by four models.

Instance	Label	m_1	m_2	m_3	m_4
x_1	1	0.70	0.60	0.00	0.65
x_2	1	0.80	1.00	1.00	0.90
x_3	1	0.80	0.95	0.93	0.88
x_4	1	0.70	0.25	0.91	0.48
x_5	0	0.80	0.68	0.78	0.74
x_6	0	0.75	0.64	0.83	0.70
x_7	0	0.10	0.37	0.78	0.24
x_8	0	0.55	0.30	0.95	0.43
x_9	0	0.80	0.72	1.00	0.76
x_{10}	0	0.15	0.25	0.87	0.20

In the general case, the loss is zero for $c \geq R(s)$. If there are l instances with score s, the previous observed positive rate is $R(s) - l/n$. For the interval $[0; R(s) - l/n]$, the loss is $Q(s, T^{rd}(c), c) = 2c$. For the interval $[R(s) - l/n; R(s)]$, the loss is derived from interpolation of the rates $R(s) - l/n$ and $R(s)$ as follows:

$$Q_0(s, T^{rd}(c), c) = 2c \left(\tfrac{R(s)-c}{R(s)-(R(s)-l/n)} \right) = 2c \left(\tfrac{R(s)-c}{l/n} \right) \tag{21}$$

When a positive rate c is desired, the instance is incorrectly classified with the frequency $\left(\frac{R(s)-c}{l/n} \right)$. The hardness of positive instances can be derived as:

$$\begin{aligned} IH_0^{rd}(s) &= \textstyle\int_0^{R(s)-l/n} 2c \,\mathrm{d}c + \int_{R(s)-l/n}^{R(s)} 2c \left(\frac{R(s)-c}{l/n} \right) \mathrm{d}c \\ &= \left[c^2 \right]_0^{R(s)-l/n} + \tfrac{2n}{l} \left[\tfrac{R(s)c^2}{2} - \tfrac{c^3}{3} \right]_{R(s)-l/n}^{R(s)} \\ &= (R(s) - l/n)^2 \tfrac{lR(s)}{n} - \tfrac{2l^2}{3n^2} = R(s)^2 + \tfrac{l}{n} \left(\tfrac{l}{3n} - R(s) \right) \end{aligned} \tag{22}$$

For large values of n, the expression approaches $R(s)^2$, which is equivalent to the continuous case (Eq. 20). In turn, Eq. 23 defines the RD cost curve for negative instances with score s and Eq. 34 the corresponding hardness measure.

$$QI_1(s,t,c) = \begin{cases} 2(1-c), \text{ if } s \leq R^{-1}(c) \\ 0, \text{ otherwise} \end{cases} \tag{23}$$

$$IH^{rd}(x) = \int_{R(s)}^{1} 2(1-c)\,\mathrm{d}c = \Big[2c - c^2\Big]_{R(s)}^{1} = (1-R(s))^2 \tag{24}$$

Hardness is given by the square of the negative rate $(1 - R(s))$. It assesses the ranking quality of the negative instances. For real datasets, the cost curve is derived by interpolating the points $R(s) - l/n$ and $R(s)$:

$$Q_1(s, T^{rd}(c), c) = 2(1-c)\left(\frac{c - R(s)}{l/n}\right) \tag{25}$$

Instance hardness is derived by Eq. 26. For large n, $IH_1^{rd}(x)$ approaches $(1 - R(s)^2)$. Fig. 4 presents the RD curve for instance x_4 using m_1. The positive rate of x_4 is $R(0.7) = 0.5$. As there are two negative instances with score 0.7, the previous rate is 0.3. The dashed area represents the interpolated loss in $[0.3; 0.5]$.

$$\begin{aligned} IH_1^{rd}(x) &= \textstyle\int_{R(s)}^{1} 2(1-c)\,\mathrm{d}c + \int_{R(s)-l/n}^{R(s)} 2(1-c)\left(\frac{c-R(s)}{l/n)}\right)\mathrm{d}c \\ &= (1-R(s))^2 + \tfrac{l}{n}\left(\tfrac{l}{3n} + (1-R(s))\right) \end{aligned} \tag{26}$$

3.4 Score-Uniform Instance Hardness

The SD method assumes that c is known at deployment and then adequate thresholds can be chosen. However, in some situations the operating condition is poorly assessed. In the worst case, a random selection is performed using the *score-uniform* (SU) method [10]: $T^{su}(c) = U[0, 1]$. The instance cost curve and hardness for a positive instance can be derived as follows:

$$\begin{aligned} QI_0(s, T^{su}(c), c) &= \textstyle\int_0^1 QI_0(s,t,c)dt \\ &= \textstyle\int_0^1 2c(1-\delta(s,t))dt \\ &= \textstyle\int_0^s 2cdt = 2cs \end{aligned} \tag{27}$$

$$IH_0^{su}(s) = \int_0^1 2csdc = s\Big[c^2\Big]_0^1 = s \tag{28}$$

The slope of the curve depends on s and ranges from 0 to $2c$ (i.e., from always correctly predicted to always incorrectly predicted). For a positive instance, $y = 0$ and then $IH_0^{su}(x) = s = |y - s|$, which is the absolute error of the model for that instance. Similarly for a negative instance, $IH_0^{su}(x) = (1 - s) = |y - s|$, again the absolute error of the model as derived below.

$$
\begin{aligned}
QI_1(s, T^{su}(c), c) &= \textstyle\int_0^1 QI_1(s, t, c)dt \\
&= \textstyle\int_0^1 2(1-c)\delta(s, t)dt \\
&= \textstyle\int_s^1 2(1-c)dt \\
&= 2(1-c)(1-s)
\end{aligned}
\tag{29}
$$

$$
\begin{aligned}
IH_1^{su}(s) &= \textstyle\int_0^1 2(1-c)(1-s)dc \\
&= (1-s)\Big[2c - c^2\Big]_0^1 = (1-s)
\end{aligned}
\tag{30}
$$

3.5 Rate-Uniform Instance Hardness

Similar to SU, uncertain operation conditions can also be defined in terms of rates. By adopting uniform distribution of positive rates, the following cost curve is derived for positive instances, with instance hardness defined in Eq. 32.

$$
\begin{aligned}
QI_0(s, T^{ru}(c), c) &= \textstyle\int_0^1 QI_0(s, R^{-1}(r), c)dr \\
&= \textstyle\int_0^1 2c(1 - \delta(s, R^{-1}(r)))dr \\
&= \textstyle\int_0^{R(s)} 2cdr = 2cR(s)
\end{aligned}
\tag{31}
$$

$$
IH_0^{ru}(x) = \int_0^1 2cR(s)dc = R(s)\Big[c^2\Big]_0^1 = R(s) \tag{32}
$$

While hardness for RD is the square positive rate, for RU it is the absolute positive rate. Poorly ranked instances will be more penalized, which is reasonable since the operation condition is uncertain. For a negative instance, hardness is its negative rate, as derived in the following equations.

$$
\begin{aligned}
QI_1(s, T^{ru}(c), c) &= \textstyle\int_0^1 QI_1(s, R^{-1}(r), c)dr \\
&= \textstyle\int_0^1 2(1-c)\delta(s, R^{-1}(r))dr \\
&= \textstyle\int_{R(s)}^1 2(1-c)dr = 2(1-c)(1-R(s))
\end{aligned}
\tag{33}
$$

$$
\begin{aligned}
IH_1^{su}(s) &= \textstyle\int_0^1 2(1-c)(1-R(s))dc \\
&= (1-R(s))\Big[2c - c^2\Big]_0^1 = (1-R(s))
\end{aligned}
\tag{34}
$$

4 Experiments

This section provides examples of the proposed cost curves and hardness measures. Fig. 5 and 6 present the cost curves respectively for the negative and positive instances in Table 1 using SF, SD and RD. The hardest negative instances

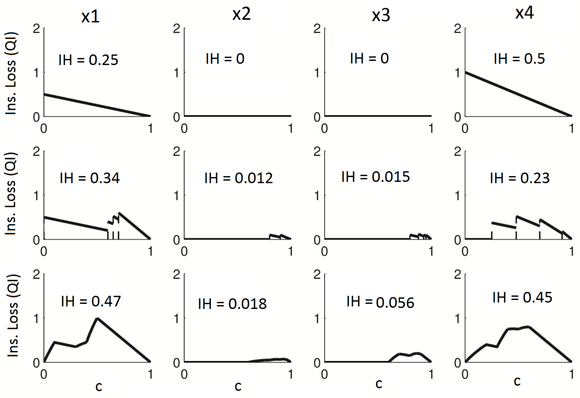

Fig. 5. Instance cost curves for negative instances considering the TCMs: SF (1st row), SD (2nd row) and RD (3rd row).

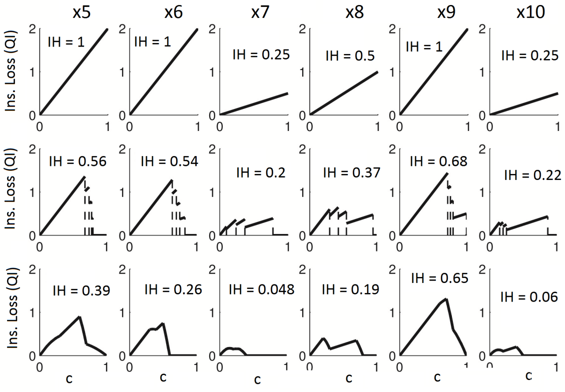

Fig. 6. Instance cost curves for positive instances considering the TCMs: SF (1st row), SD (2nd row) and RD (3rd row).

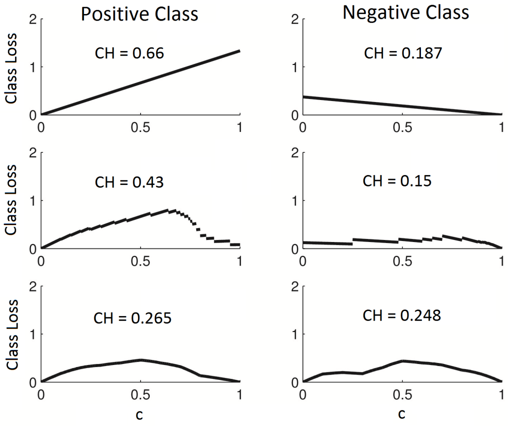

Fig. 7. Class cost curves and hardness under different TCMs.

are x_1 and x_4. In particular, x_1 is even harder to rank, given the RD hardness. Considering the positive class, x_5, x_6 and x_9 have the highest hardness values. However, for higher costs, they are easy for RD and SD. Different from SF, the RD and SD methods can take advantage on the operation condition known in deployment. Fig. 7 in turn presents the *class cost curves* produced by averaging the instance cost curves for each class. Class hardness (CH) is defined as the average instance hardness in a given class. It is an estimation of the class loss defined in Eq. 7 and 10. By assuming SF and SD, the positive class is relatively more difficult than the negative class. A more balanced difficulty is observed by assuming the RD method. Although the scores of the positive instances are not well calibrated, they can produce a good ranking of instances.

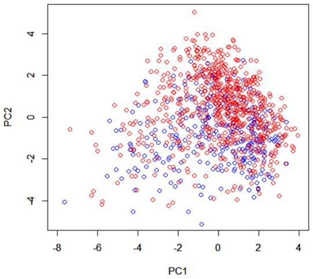

Fig. 8. German dataset visualized using PCA.

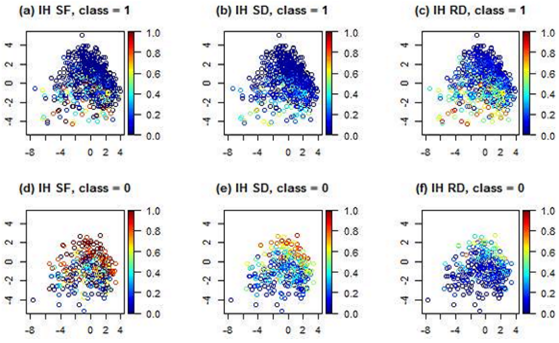

Fig. 9. Hardness of instances for the German-Credit dataset.

Following, the framework was applied to a real dataset (German Credit, in Fig. 8). The negative class is the majority (700 instances), while the positive class has 300 instances. Both classes are largely spread, although the negative class seems to be more compact. There is a class boundary in which the classes are highly mixed. Five models were learned in this dataset using diverse algorithms in Weka[2], with scores computed by 10-fold cross validation. Scores were more concentrated towards 1, as negative is the majority class. By considering a fixed threshold 0.5, many errors were observed for the positive class, particularly in the class boundary (see Fig. 9(d)). The negative class is much easier (class hardness is 0.12 against 0.56 for the positive class). By considering SD, as thresholds are adapted, instances are in general easier, compared to SF (see Fig. 9(b) and (e)). In fact, positive class hardness is 0.37 for SD. As there are still some hard positive instances in the boundary, this class is still much harder than the negative one (whose hardness is 0.10). For RD, hardness is more balanced among classes. Some negative instances are poorly ranked (see Fig. 9(c)). On the other hand, some positive instances in the boundary, which are difficult for SF and SD, are easier to rank (see Fig. 9(f)). For RD, class hardness is respectively 0.25 and 0.17 for negatives and positives. The negative class becomes harder than the positive. Although with good absolute scores, the negative instances are harder to rank.

Differences in difficulty can also be analyzed at specific operation conditions. For the negative class, higher losses tend to be observed for higher values of c, as expected. However different patterns are seen depending on the TCM (see Fig. 10). For $c = 0.8$, the number of hard instances for RD is high, but extremely hard instances are not observed. Notice that false positives are penalized by a low cost in this case $(1 - c) = 0.2$. For $c = 0.5$, in turn, some very hard instances in the class boundary are observed for RD. Distinct patterns can also be observed for the positive class, which is difficult for SD (see Fig. 11). For $c = 0.2$ in SD,

[2] J48, IBk, Logistic Regression, Naive Bayes and Random Forest were adopted. IBK adopted k=5. The other algorithms were applied using default parameter values.

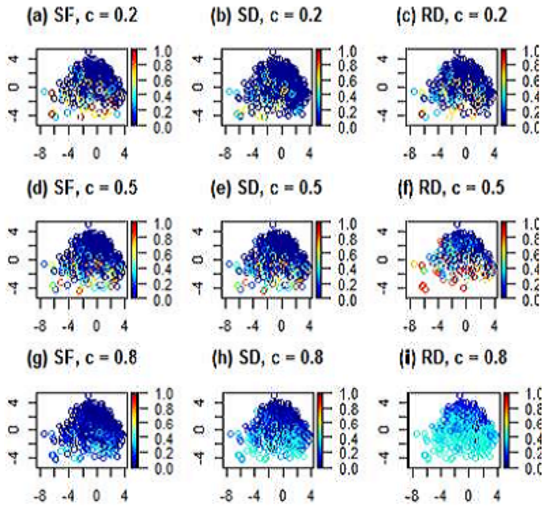

Fig. 10. Instance hardness for different c - Class 1.

most instances are hard, but not extremely hard as for $c = 0.5$. In this case, the higher cost impacts instance hardness.

Fig. 12 presents the instance hardness for SU and RU, in which c is uncertain. In these cases, hardness is more distributed and more difficult instances are found beyond class boundary. Class hardness for SU is 0.22 and 0.54 respectively for classes 1 and 0, which represents a harder scenario compared to SD. Similarly for RU, class hardness is 0.39 and 0.35, which is greater than class hardness for RD. The increase in hardness reflects the uncertainty in the cost proportions.

5 Conclusion

This paper proposes a new framework for measuring instance hardness in binary classification. This work addresses different perspectives of evaluation by considering different TCMs in the definition of instance hardness. Future works point at three directions: (1) derive new measures within the framework by adopting other TCMs and distributions of operating condition; (2) perform more extensive experiments on a large set of real problems using the proposed measures - such studies would reveal advantages, limitations and relationships between algorithms in different scenarios, which is relevant for understanding learning behavior [2]; and (3) develop applications in different contexts. In *dynamic algorithm selection*, for example, instance cost curves can be adopted to select algorithms for specific regions in the instance space given the operation condition. In *active learning*, expected hardness can be used for selecting unlabeled instances for label acquisition. In *noise filtering* and *acquisition of missing values*, the effect of data preprocessing in the instance hardness can be analyzed.

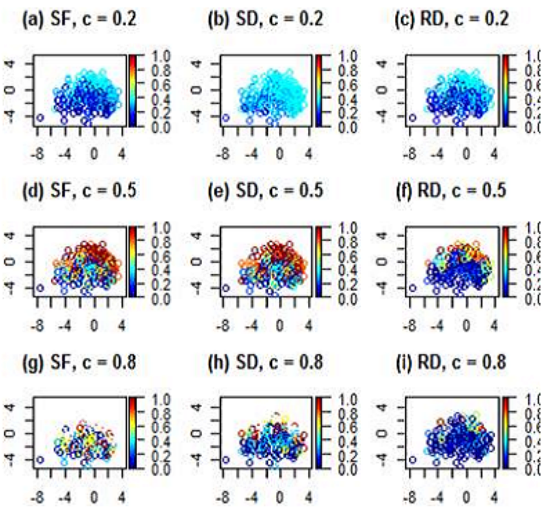

Fig. 11. Instance hardness for different c - Class 0.

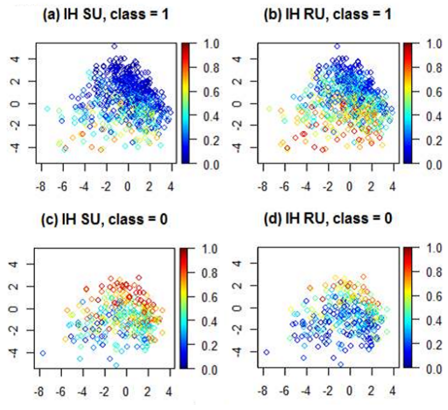

Fig. 12. Instance hardness under the SU and RU methods.

References

1. Basu, M., Ho, T. (eds.): Data complexity in pattern recognition. Springer (2006)
2. Brazdil, P., Giraud-Carrier, C.: Metalearning and algorithm selection: progress, state of the art and introduction to the 2018 special issue. Machine Learning **107**(1), 1–14 (2018)
3. Brodley, C.E., Friedl, M.A.: Identifying mislabeled training data. Journal of Artificial Intelligence Research **11**, 131–167 (1999)
4. Cruz, R., Sabourin, R., Cavalcanti, G.: Prototype selection for dynamic classifier and ensemble selection. Neural Computing and Applications **29**(2), 447–457 (2018)
5. Drummond, C., Holte, R.C.: Cost curves: An improved method for visualizing classifier performance. Machine Learning **65**(1), 95–130 (2006)
6. Flach, P., Matsubara, E.T.: A simple lexicographic ranker and probability estimator. In: ECML 2017. pp. 575–582 (2007)
7. Garcia, L.P., Carvalho, A.C., Lorena, A.C.: Effect of label noise in the complexity of classification problems. Neurocomputing **160**, 108 – 119 (2015)
8. Hernández-Orallo, J., Flach, P., Ferri, C.: Brier curves: A new cost-based visualisation of classifier performance. In: 28th Intern. Conf. on Machine Learning (2011)
9. Hernández-Orallo, J., Flach, P., Ferri, C.: A unified view of performance metrics: Translating threshold choice into expected classification loss. Journal of Machine Learning Research **13**(1), 2813–2869 (2012)
10. Hernández-Orallo, J., Flach, P., Ferri, C.: Roc curves in cost space. Machine Learning **93**(1), 71–91 (2013)
11. Luengo, J., Shim, S.O., Alshomrani, S., Altalhi, A., Herrera, F.: Cnc-nos: Class noise cleaning by ensemble filtering and noise scoring. Knowledge-Based Systems **140**, 27 – 49 (2018)
12. Martınez-Plumed, F., Prudêncio, R.B., Martınez-Usó, A., Hernández-Orallo, J.: Making sense of item response theory in machine learning. In: European Conference on Artificial Intelligence, ECAI. pp. 1140–1148 (2016)
13. Melville, P., Mooney, R.J.: Diverse ensembles for active learning. In: Proceedings of the 21st International Conference on Machine Learning. p. 74 (2004)
14. Morán-Fernández, L., Bolón-Canedo, V., Alonso-Betanzos, A.: Can classification performance be predicted by complexity measures? a study using microarray data. Knowledge and Information Systems **51**(3), 1067–1090 (2017)
15. Napierala, K., Stefanowski, J.: Types of minority class examples and their influence on learning classifiers from imbalanced data. Journal of Intelligent Information Systems **46**(3), 563–597 (2016)
16. Sluban, B., Lavrac, N.: Relating ensemble diversity and performance: A study in class noise detection. Neurocomputing **160**, 120 – 131 (2015)
17. Smith, M.R., Martinez, T., Giraud-Carrier, C.: An instance level analysis of data complexity. Machine Learning **95**(2), 225–256 (2013)
18. Verbaeten, S., Assche, A.V.: Ensemble methods for noise elimination in classification problems. In: Proc. 4th Int. Conf. Multiple Classifier Syst. pp. 317–325 (2003)
19. Woloszynski, T., Kurzynski, M., Podsiadlo, P., Stachowiak, G.W.: A measure of competence based on random classification for dynamic ensemble selection. Information Fusion **13**(3), 207–213 (2012)
20. Woods, K., Kegelmeyer, W., Bowyer, K.: Combination of multiple classifiers using local accuracy estimates. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**, 405–410 (1997)