Technology
Arts Sciences
TH Köln

Approximating the optimal threshold for an abstaining classifier based on a reward function with regression

Bachelor Thesis

Jonas Fassbender jonas@fassbender.dev
11117674

In the course of studies
Computer Science

For the degree of
Bachelor of Science

Technical University of Cologne
Faculty of Computer Science and Engineering

First supervisor: Prof. Dr. Heinrich Klocke
Technical University of Cologne

Second supervisor: Prof. Dr. Fotios Giannakopoulos
Technical University of Cologne

Overath, July 2019

1. Introduction

An abstaining classifier (see e.g. Vanderlooy et al., 2009)—also called a classifier with reject option (see e.g. Fischer et al., 2016)—is a kind of confidence predictor. It can refuse from making a prediction if its confidence in the prediction is not high enough. High enough, in this context, means that the confidence is greater than a certain—hopefully optimal—threshold. Optimality is dependent on a performance metric set beforehand.

This thesis introduces a new kind of method for approximating the optimal threshold based on a reward function—better known from reinforcement learning than from the supervised learning setting (see e.g. Sutton and Barto, 2018, Chapter 1). The method treats the reward function as unknown, making it a very general approach and giving quite the amount of freedom in designing the reward function.

In supervised learning the concept that is closest to a reward function is a cost function and many abstract types of cost in supervised learning are known (see Turney, 2002).

Probably today's most used methods for obtaining the optimal threshold for reducing the expected cost of an abstaining classifier are based on the receiver operating characteristic (ROC) rule (see Tortorella, 2000; Pietraszek, 2005; Vanderlooy et al., 2009; Guan et al., 2018).

The method presented in this thesis is more flexible than the methods based on the ROC rule and can—depending on the context of the classification problem—produce results better interpretable than results from a cost setting (see Chapter 2). Also it is more natural with multi-class classification problems than the methods based on the ROC rule, all assuming binary classification problems, wherefore the classifiers generated by these methods must be transformed to multi-class classifiers for non-binary problems.

On the other hand the presented method can suffer from its very general approach and only produces approximations. This can result in non-optimal and unstable thresholds.

This thesis first presents a motivational example. In Chapter 3 the proposed method is presented. After that experiments on data sets from the UCI machine learning repository (see Dua and Graff, 2017) are discussed. At last further research ideas are listed and a conclusion is drawn.

2. Motivational example

This chapter will point out the usefulness of abstaining classifiers in real world application domains where reliability is key. It will show an example why the reward setting can improve readability in some domains. First another example, for which the cost setting—more commonly used in supervised learning—comes more natural is given and the differences are discussed.

Abstaining classifiers—compared to typical classifiers, which classify every prediction, maybe even without a confidence value in it (then called a bare prediction)—can be easily integrated into and enhance processes where they partially replace some of the decision

making, since they can delegate the abstained predictions back to the underlying process. The use of abstaining classifiers in domains where reliability—in regard to prediction errors—is important, has an interesting aspect in giving reliability while still being able to decrease work, cost, etc. to some degree. This is a valuable property if there does not exist a typical classifier good enough to fully replace the underlying process.

Many real world application domains for abstaining classifiers can express a cost function associated to the decisions about predicting and abstaining of the classifier—which then chooses the threshold with which it produces the least amount of cost, therefore minimizing the cost of introducing the abstaining classifier to the process.

For example, the real world application domain could be a facial recognition system at a company which regulates which employee can enter a trust zone and which can not. The process which should be enhanced with the facial recognition system is a manual process where the employee has to fill out a form in order to receive a key which opens the trust zone.

In this example, the costs of miss-classifying an unauthorized person as authorized can be huge for the company while abstaining or classifying an authorized employee as unauthorized produces quite low costs—the authorized employee just has to start the manual process, which should be replaced by the facial recognition system.

On the other hand, for some real world application domains a reward function based on which the abstaining classifier chooses the threshold by maximizing the reward—rather than minimizing the cost—comes more natural.

Such a domain would be the finance industry, where we often can associate a certain amount of money an abstaining classifier can produce or safe by supporting the decision making of an underlying process.

An example for such a process would be the process of a bank for granting a consumer credit. The bank requests information about the consumer from a credit bureau in order to assess the consumer's credit default risk. Now the bank wants to predict the consumer's credit default risk based on information the bank has about the consumer. If the credit default risk is very high or very low the bank can save money not making a request to the credit bureau for this consumer. The optimal threshold for the abstaining classifier making the prediction about the credit default risk can easily be expressed by a reward function. Every correct decision saves the bank the money the request to the credit bureau costs. Every miss-classification costs the bank either the amount of money it would gain by granting the credit, or the money it loses by giving a credit to somebody that does not pay the rates. Abstention cost is the cost of making a request to the credit bureau.

Using a reward function—like in the example above—instead of a cost function has an advantage in readability. One can easily assess the gain of introducing the abstaining classifier to the process. Is the reward generated by the abstaining classifier higher than zero, the process is enhanced by the abstaining classifier. Otherwise the abstaining classifier would produce more cost than it would save and it is not valuable for the bank to introduce it to its process of assessing a consumer's credit default risk.

3. Proposed method based on reward

Let $\mathbf{X}$ be the observation space and $\mathbf{Y}$ the label space. $|\mathbf{Y}| < \infty$ since only classification is discussed. Let $\mathbf{Z}$ be the cartesian product of $\mathbf{X}$ and $\mathbf{Y}$: $\mathbf{Z} := \mathbf{X} \times \mathbf{Y}$. $\mathbf{Z}$ is called the example space. Let an example z_i from $\mathbf{Z}$ be: $z_i := (x_i, y_i); z_i \in \mathbf{Z}$. A data set[1] containing examples $z_1, \ldots, z_n$ is annotated as $\wr z_1, \ldots, z_n \wr$.

3.1 Scoring classifiers

A classical machine learning predictor—in the previous chapter called a typical classifier—can be represented by a function

$$D : \mathbf{Z}^* \times \mathbf{X} \to \mathbf{Y}. \tag{1}$$

Its first argument being a data set with an arbitrary length the classifier is trained on, while the second is an observation which should be predicted (mapped to a label from $\mathbf{Y}$).

Let $D_{\wr z_1,\ldots,z_n \wr}$ be a classical machine learning predictor trained on the data set $\wr z_1, \ldots, z_n \wr$ and let $D_{\wr z_1,\ldots,z_n \wr}(x)$ be equivalent to (1) with the first argument being $\wr z_1, \ldots, z_n \wr$.

The proposed method relies on scoring classifiers. A scoring classifier does not return just a label but instead returns some score for each label from the label space. The only constraint on the scores is that higher scores are better than lower. A score could be a probability or just an uncalibrated confidence value (see Vanderlooy et al., 2009).

Let S be a scoring classifier:

$$S : \mathbf{Z}^* \times \mathbf{X} \to (\mathbf{Y} \to \mathbb{R}). \tag{2}$$

S takes the same arguments as (1) but instead of producing bare predictions it returns a function which maps every label from the label space to a score determined by S.

The method proposed is only interested in the highest score and the associated label. For that two functions k and v are defined:

$$k(S_{\wr z_1,\ldots,z_n \wr}, x) = \arg\max_{y \in \mathbf{Y}} S_{\wr z_1,\ldots,z_n \wr}(x)(y)$$
$$v(S_{\wr z_1,\ldots,z_n \wr}, x) = \max_{y \in \mathbf{Y}} S_{\wr z_1,\ldots,z_n \wr}(x)(y).$$

The composition kv of k and v returns the tuple with the label mapped to the highest score:

$$kv(S_{\wr z_1,\ldots,z_n \wr}, x) = (k(S_{\wr z_1,\ldots,z_n \wr}, x), v(S_{\wr z_1,\ldots,z_n \wr}, x)). \tag{3}$$

1. not an actual set but a multi-set since it can contain the same element more often than one time.

3.2 Abstaining classifiers

An abstaining classifier A can be defined as a similar function as (1), with the only difference being the return value:

$$A : \mathbf{Z} \times \mathbf{X} \to \mathbf{Y} \cup \{\perp\}$$

A can return a label from $\mathbf{Y}$, but also $\perp$, indicating that A would like to abstain from making a prediction.

Let $\mathbf{S_Z}$ be the set of all scoring classifiers defined like (2) on the example set $\mathbf{Z}$. The proposed method is interested in transforming a scoring classifier $S \in \mathbf{S_Z}$ to an abstaining classifier A. In order to do that a threshold $T \in \mathbb{R}$ is defined and A can be represented as a composition of S and T. Let $S_{\wr} z_1, \ldots, z_n \wr$ be a scoring classifier, T a threshold and x an observation to be predicted. The abstaining classifier A composed of $S_{\wr z_1,\ldots,z_n \wr}$ and T predicts x as follows:

$$A(\wr z_1, \ldots, z_n \wr, x) = \begin{cases} k(S_{\wr z_1,\ldots,z_n \wr}, x) & \text{if } v(S_{\wr z_1,\ldots,z_n \wr}, x) \geq T \\ \perp & \text{if } v(S_{\wr z_1,\ldots,z_n \wr}, x) < T \end{cases}. \tag{4}$$

This representation of A is rather unconventional and is one reason the proposed method is unstable.

Using a single threshold for all labels is a strong constraint to put onto the scoring classifier, because it must be invariant to the label distribution. Imagine a classification problem where one label makes up 90 percent of all examples and the scoring classifier is not invariant to the label distribution. This could lead the classifier to produce higher scores for observations with the label which makes up 90 percent. This could result in an abstaining classifier that does not predict an any example which does not have the dominant label, even though with such a distribution predicting the submissive labels would probably be more interesting.

ROC based and other methods for generating abstaining classifiers address this problem by using abstention windows instead of a single threshold (see Friedel et al., 2006).

Let $\mathbf{Y}$ be a binary problem $\mathbf{Y} := \{P, N\}$, where P is called the positive label and N the negative label. The margin $m : \mathbf{Y} \times \mathbb{R} \to (-1, 1)$ is a function that combines the label with the confidence value and returns a number in the interval of $(-1, 1)$. The closer the return value of m is to the edges of the interval, the more confident the scoring classifier is, whereby -1 means perfectly confident the label is N and 1 means perfectly confident the label is P (see Friedel et al., 2006).

In Guan et al. (2018) a similar method is described, constraining the output of the margin m not on $(-1, 1)$ but instead using only the likelihood of an observation x having the positive label P ($m : \mathbb{R} \to (0, 1)$).

Both Friedel et al. (2006) and Guan et al. (2018) define an abstention window a as a tuple $a := (t_1, t_2); t_1 < t_2$ with two thresholds. An abstaining classifier of the form

described in (4) with an abstention window instead of a threshold predicts an observation x as:

$$A(\wr z_1, \ldots, z_n \wr, x) = \begin{cases} P & \text{if } m(kv(S_{\wr z_1,\ldots,z_n \wr}, x)) > t_2 \\ \perp & \text{if } t_1 \leq m(kv(S_{\wr z_1,\ldots,z_n \wr}, x)) \leq t_2 \\ N & \text{if } m(kv(S_{\wr z_1,\ldots,z_n \wr}, x)) < t_1 \end{cases}.$$

This addresses the problem of using a single threshold T for predictions on both labels from $\mathbf{Y}$. The constraint of abstention windows is that they are only defined on binary problems and must be transformed in order to use them in a multi-class setting. This could be done with the one-vs-one or the one-vs-all approach, in which multiple binary classifiers are learned (see e.g. Murphy, 2012, Chapter 14.5). But, like stated in Friedel (2005) multi-class problems increase the complexity of ROC based and other methods, because when using a one-vs-one or one-vs-all approach it is possible that more than one label gets predicted by the abstaining classifier (see Friedel, 2005).

On the other hand an arbitrary number of labels can be predicted with a single threshold, though the solution could be sub-optimal and is depending heavily on the underlying scoring classifier.

This thesis does not address the problem of using a single threshold in the empirical study presented in Chapter 4, but a possible solution is given in Chapter 5.

3.3 Abstaining classifiers based on a reward system

The novel approach of this thesis is using a system based on reward which is maximized rather then cost that is minimized in order to determine the optimal threshold for abstention. Like stated in Chapter 1 using a reward function—like used in reinforcement learning—in a supervised learning setting is rather uncommon. In Chapter 1 and Chapter 2 some reasons why using reward instead of cost are given.

Another aspect of cost, which makes it less flexible than reward, not previously discussed, is that it is only defined on $\mathbb{R}^+$, while reward is defined on $\mathbb{R}$. Reward combines cost with gain.

Let ρ be a reward function:

$$\rho : \mathbf{Y}^* \times \hat{\mathbf{Y}}^* \to \mathbb{R}^*. \tag{5}$$

ρ takes two arbitrary, but equal long vectors with labels from $\mathbf{Y}$ and from $\hat{\mathbf{Y}}$. $\hat{\mathbf{Y}}$ can be equal to $\mathbf{Y}$ or also contain an element indicating abstention $\perp$. The first vector contains the true labels of some sequence of examples, the second contains the predicted labels from some classifier for the same sequence. ρ returns a reward for each tuple of true label and predicted label from the parameter vectors.

The reward function is basically treated as a black box function; the only knowledge we have is, whether ρ produces single-step reward or accumulated reward values and whether ρ is stateful or stateless (see Table 1).

	stateless	stateful
accumulated	true	true
single step	true	false

Table 1: Possible combinations of the two known properties of a reward function. It is not possible to have a stateful single step reward function, because a single step reward function is only dependent on the true and predicted label of one example.

Treating the reward function this way makes it more flexible than a cost setting which uses cost matrices (see Fischer et al., 2016). A cost matrix C for a binary abstaining classifier is defined as

$$C := \begin{pmatrix} C(P,P) & C(P,N) & C(P,\bot) \\ C(N,P) & C(N,N) & C(N,\bot) \end{pmatrix}.$$

A cost function c with the same definition as (5) based on such a cost matrix C would be defined as $c(\vec{t},\vec{p}) = [C(t_i,p_i); i = 1,\ldots,|\vec{t}|]^T$ and is basically the inverse of a single step reward function—with the difference that $C(P,P)$ and $C(N,N)$ normally do not have a gain associated to them, because then the cost matrix would not be true to its cost setting. A cell of a cost matrix C would provide a gain if its value is smaller than zero.

A reward function that returns already accumulated rewards provides an even more flexible setting than single step reward—which is only dependent on one example's true and predicted label—because it can introduce the concept of state (see Sutton and Barto, 2018, Chapter 1).

For example, the abstaining classifier could be a bettor betting on the outcome of a card game. It starts with a certain amount of money and always bets two thirds of the amount it currently has. Every example is one match and it is possible to derive a certainty measure based on some information about the match. The reward is the amount of money the classifier wins or loses. It gains a certain amount—depending on how much money the classifier owns after the last match it has bet on—if it decides to bet on the current match and does so correctly or loses two thirds of its reward up to the current bet if the classifier was wrong. A reward function like this is not stateless like a single step reward function and is a commonly used in the reinforcement learning setting (see Sutton and Barto, 2018, Chapter 1).

An interesting question is where to draw the line between an abstaining classifier that maximizes reward and a reinforcement learning agent, because the bettor described above could also be defined as a reinforcement learning agent. This thesis will not declare a clear differentiation between the two concepts, but the interaction with the environment seems to be a good point for differentiation. If the predictions of the abstaining classifier alter

reality (the predictions of the bettor above most certainly would change reality) it behaves like an agent, otherwise it is just an abstaining classifier.

An argument for such a differentiation would be, that supervised learning—on which the focus of this thesis lies—is underlined by the assumption, that all observations $x_i \in \mathbf{X}$ observed are independent from the other observations $x_j \in \mathbf{X}$, but that they share the same unknown distribution. This assumption is called the iid assumption (independent and identically distributed) (see Clauset, 2011) and makes the concept of state irrelevant to our observations, which would not be the case if the predictions alter reality.

3.4 Method for approximating the optimal threshold for abstention based on a reward function

For approximating the optimal threshold—which maximizes the expected reward in the proposed reward setting—an architecture comparable to and influenced by the meta-conformal prediction approach described in Smirnov et al. (2009) is proposed. The architecture of an abstaining classifier based on reward is comparable to the combined classifier used for meta-conformal prediction. A combined classifier *B:M* uses a base classifier B defined like (1) and a conformal predictor M in order to extend B with a confidence measure. *B:M* can then be transformed to an abstaining classifier by defining a threshold T in the confidence values generated by M using the ROC isometrics approach (see Smirnov et al., 2009; Vanderlooy et al., 2009; Fassbender, 2019).

An abstaining classifier A in the reward based setting described above can also be described as a combined classifier $A := S{:}Reg$. S is a scoring classifier defined like (2) and *Reg* is a regressor (defined like (1) with $\mathbf{Y} := \mathbb{R}$). *Reg* is not necessarily needed and is only used in order to determine the threshold T for A, which can be done by other means. Chapter 4 shows how well using different regressors perform in comparison to just taking the threshold which has generated the highest reward on the training set.

The threshold of A that approximates the maximum expected reward is defined during the training phase. Let $(z_1, \ldots, z_n)$ be a training set. $(z_1, \ldots, z_n)$ is split into k roughly equal sized partitions using the k-fold method (see Hastie et al., 2009, Chapter 7.10; Algorithm 1, line 2).

For each partition combine the other $k-1$ partitions to a training set; train a scoring classifier S on this set and let it predict on the partition it was not trained on. Add the true labels from the examples in the predicted partition and kv (see Equation 3) of all predictions to a prediction set $P \subseteq \mathbf{Y}^n \times \hat{Y}^n \times \mathbb{R}^n$ (see Algorithm 1, lines 3–11).

After that P is sorted in descending order based on the scores, transforming it into a sequence. The reward from the first two columns of P—with a reward function ρ defined like (5)—is computed. Since iid (see previous chapter) is assumed, the parameter vectors for ρ can be provided with any ordering and stateful reward functions must also assume iid on the sequence it sees as its parameters. Every reward is related to an element from P and the reward is combined with the scores from P to build the reward points $R :=$

Algorithm 1 : k-fold method for determining the threshold for an abstaining classifier based on a reward function

Input:

S: a socring classifier,
ρ: a stateless reward function defined like (5),
data set: $\wr z_1, \ldots, z_n \wr$,
k: the amount of partitions,
Reg: a regressor (optional)

Output:

T: threshold

$P := \{\}$
split data set into k roughly equal sized partitions $\text{split}_1, \ldots, \text{split}_k$
for all $\text{split}_i, i = 1, \ldots, k$ **do**
    combine all splits $\neq \text{split}_i$ to a training set
    train S with the training set
    let S predict examples in split_i
    for all elements in prediction of $S \times$ the true labels of split_i **do**
        get the label associated with the highest score for the element with (3)
        add the true label, the predicted label and the score to P
    end for
end for
sort P based on the scores in descending order
$R :=$ scores from $P \times \rho(P)$
if ρ is a single step reward function **then**
    accumulate reward in R
end if
reduce R so all scores are unique (optional)
train *Reg* (optional)
determine T with (6) or (7)
return T

s_i	r_i		s_i	r_i
0.98	1			
0.98	2		0.98	2
0.97	3			
0.97	2	$\Rightarrow$		
0.97	1		0.97	1
0.96	0		0.96	0

Figure 1: An example for reducing R built from a single-step reward function which gives $+1$ for a correct prediction and -1 for a false one. R is already accumulated. If T would be determined from the unreduced R, 0.97 would be the optimal threshold, because it is $\arg\max_{r_i} R$. The problem is that two errors produced by the scoring classifier for examples with the same certainty 0.97 are concealed. Reducing R to only the last element of each score makes certain that no concealing can happen.

$[(s_i, r_i); i = 1, \ldots, n]; R \in \mathbb{R}^{n \times 2}$, where the scores are mapped to their associated rewards (see Algorithm 1, lines 12, 13).

If ρ is a single step reward function the rewards are accumulated, which only means that the reward at a single point is the sum of all rewards previously seen (all previously seen scores are higher or equal to the score of the single point, since R is a sorted sequence). The accumulated version of R is $R' := [(s_i, \sum_{j=1}^{i} r_j); i = 1, \ldots, n]$ (see Algorithm 1, lines 14–16).

At last T is derived from R. If Reg is defined it is trained on R, with s_i as observation and r_i as the label. T is set equal to the score for which Reg predicts the highest reward; the local maximum of Reg. Only the local maximum is of interest, which means T must lie in the interval derived from the convex hull of all scores generated from S during the training:

$$T := \underset{s \in \text{convex hull}(s_i \in R)}{\arg\max} Reg(s). \tag{6}$$

If Reg is not defined T could be derived from R by taking the score which has the highest associated reward

$$T := \underset{s_i}{\arg\max}\, R. \tag{7}$$

For determining T like this, R would be reduced so each score is unique, since T can only split between two scores s_i, s_j, if $s_i \neq s_j$. This step is optional (see Algorithm 1, line 17).

Making R unique could be done in different ways, for example—if ρ is a single step reward function—it would make sense to take the last tuple of a sub-sequence where each tuple has the same score, since it contains the most information about the reward (see Figure 1). One could also reduce them by averaging their rewards, etc.

3.5 Equivalences to reinforcement learning

Using a reward system—like described above—to determine an abstaining classifier makes the whole process quite similar to the whole setting of reinforcement learning; not only the reward part. This chapter lists some more aspects which makes a reinforcement learning agent and an abstaining classifier look alike, but also shows where both concepts differ.

A reinforcement learning agent, also called the autonomous agent, observes—for each (time-)step t—a state s_t from its environment. Based on s_t the autonomous agent takes an action a_t and the environment transitions to a new state s_{t+1}. At each transition the environment provides a reward value; a feedback for the agent on how well it performs. The agent learns a policy with which it maximizes the expected reward (see Arulkumaran et al., 2017).

An abstaining classifier works quite the same way. It observes an observation $x_t \in \mathbf{X}$. For x_t it produces a prediction $p_t \in \mathbf{Y} \cup \{\bot\}$.

The reward system works a little differently than the one used in reinforcement learning. Assuming a direct reward would mean that the abstaining classifier is used in the perfect online setting in which reality provides the correct answer after every prediction, which is seldom the case and would make the abstaining classifier redundant (see Vovk et al., 2005, Chapter 4.3). While the autonomous agent can use trial and error in order to increase the success of its policy, the abstaining classifier is bound to the already observed data and can only try to generalize from the previous observations to unseen ones. The equivalent of the abstaining classifier to the policy of the agent would be its threshold T (see Table 2).

The most obvious difference is that the agent actively interacts with the environment, while the abstaining classifier should be irrelevant to its environment—reality providing the classifier with examples but without assuming the predictions in any case alter the environment, because it would violate the iid assumption. The only way interaction with the environment can be indirectly represented is through a stateful reward function, which can simulate decisions made by the abstaining classifier (see Chapter 3.3).

4. Experiments

This chapter will show some experiments in which the proposed method, with different configurations, is tested on real-world data sets from the UCI machine learning repository (see Dua and Graff, 2017).

Reinforcement learning agent	Abstaining classifier
state s_i	observation x_i
action a_i	prediction p_i
environment	reality providing examples from $\mathbf{Z}$
action changes state of environment	iid assumption (reality not altered by predictions)
policy π	threshold T
trial and error	examples from reality

Table 2: Comparison of a reinforcement learning agent with an abstaining classifier in the reward setting.

The configuration contains two different scoring classifiers, eighteen reward functions and 5 regressors—approximating the optimal threshold like (6)—plus the bare threshold derived like (7).

4.1 Data sets

Six data sets where chosen for the experiments. The first being the bank marketing data set[2] (`bank`). This data set contains information about the success of a marketing campaign (phone calls) of a Portuguese bank. The goal is to predict whether a phone call to a potential customer results in success, which means the potential customer subscribes to a term deposit (see Moro et al., 2014). The data set has seventeen features and is a binary classification problem with 41,188 examples. The data set is unbalanced, it contains far less successful phone calls then unsuccessful ones (see Moro et al., 2014).

The second data set tested was `bank-additional`. It is the same data set as `bank`, but has three more features.

The third data set is the car evaluation data set[3] (`car`). It is described in Bohanec and Rajkovič (1988) and contains 1,728 examples with six attributes. Noteworthy is the fact that all features and the label are discrete with just three or four manifestations.

Also tested where the default of credit card clients data set[4] (`credit card`). It contains information about default payments in Taiwan and the goal is to predict whether an observation represents a credible client or not (see Yeh and hui Lien, 2009). This is closely related to the example described in Chapter 2.

Probably the most famous data set used is USPS data set (`usps`). It is used in hundreds of papers and books. It contains 9,298 examples (images) of handwritten digits from real life zip codes collected by the US Postal Service office in Buffalo, NY (see Vovk et al., 2005,

2. `https://archive.ics.uci.edu/ml/datasets/bank+marketing`
3. `https://archive.ics.uci.edu/ml/datasets/Car+Evaluation`
4. `https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients`

data set	# examples	# features	$\lvert\mathbf{Y}\rvert$
`bank`	41,188	17	2
`bank-additional`	41,188	20	2
`car`	1,728	6	4
`credit card`	30,000	24	2
`usps`	9,298	256	10
`wine`	6,497	12	11

Table 3: Characteristics of the tested data sets.

Appendix B.1). The observations are a 16×16 matrix where each cell is in the interval of $(-1, 1)$. Each cell represents the brightness of a pixel. The labels are the interval 0 to 9 (see LeCun et al., 1989; Fassbender, 2019).

The last data set tested was the wine quality data set[5] (`wine`). Each example represents a sample of "vinho verde" from northern Portuguese. The twelve attributes are physicochemical properties of the sampled wine which are mapped to a sensory output—the quality of the wine from zero to ten. The data set is unbalanced in two ways. It contains only 1,599 red wine samples, but 4,898 white wine samples. The more important imbalance is the distribution of the label. Most samples have a quality of five or six (see Cortez et al., 2009).

4.2 Scoring classifiers

4.3 Reward functions

4.4 Regressors

4.5 Results

5. Further research

6. Conclusion

Appendix

A. Plots

References

Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. A brief survey of deep reinforcement learning. *CoRR*, abs/1708.05866, 2017. URL

5. `https://archive.ics.uci.edu/ml/datasets/Wine+Quality`

http://arxiv.org/abs/1708.05866.

Marko Bohanec and Vladislav Rajkovič. V.: Knowledge acquisition and explanation for multi-attribute decision. In *Making, 8 th International Workshop "Expert Systems and Their Applications*, 1988.

Aaron Clauset. Inference, models and simulation for complex systems: A brief primer on probability distributions. 08 2011. URL http://tuvalu.santafe.edu/~aaronc/courses/7000/csci7000-001_2011_L0.pdf.

Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547 – 553, 2009. ISSN 0167-9236. doi: https://doi.org/10.1016/j.dss.2009.05.016. URL http://www.sciencedirect.com/science/article/pii/S0167923609001377. Smart Business Networks: Concepts and Empirical Evidence.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Jonas Fassbender. libconform v0. 1.0: a python library for conformal prediction. *arXiv preprint arXiv:1907.02015*, 2019. URL https://arxiv.org/abs/1907.02015.

Lydia Fischer, Barbara Hammer, and Heiko Wersing. Optimal local rejection for classifiers. *Neurocomputing*, 214:445 – 457, 2016. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2016.06.038. URL http://www.sciencedirect.com/science/article/pii/S0925231216306762.

Caroline Friedel. On abstaining classifiers. 01 2005.

Caroline C. Friedel, Ulrich Rückert, and Stefan Kramer. Cost curves for abstaining classifiers. In *Proceedings of the ICML 2006 workshop on ROC Analysis in Machine Learning*, 2006.

Hongjiao Guan, Yingtao Zhang, Heng-Da Cheng, and Xianglong Tang. Abstaining classification when error costs are unequal and unknown. *CoRR*, abs/1806.03445, 2018. URL http://arxiv.org/abs/1806.03445.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, second edition, 2009.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, December 1989. ISSN 0899-7667. doi: 10.1162/neco.1989.1.4.541. URL http://dx.doi.org/10.1162/neco.1989.1.4.541.

Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22 – 31, 2014. ISSN 0167-9236. doi: https://doi.org/10.1016/j.dss.2014.03.001. URL http://www.sciencedirect.com/science/article/pii/S016792361400061X.

Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN 0262018020, 9780262018029.

Tadeusz Pietraszek. Optimizing abstaining classifiers using roc analysis. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 665–672, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. doi: 10.1145/1102351.1102435. URL http://doi.acm.org/10.1145/1102351.1102435.

Evgueni Smirnov, Georgi Nalbantovi, and A. M. Kaptein. Meta-conformity approach to reliable classification. *Intelligent Data Analysis*, 13, 01 2009. doi: 10.3233/IDA-2009-0400.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 2nd edition, 2018.

Francesco Tortorella. An optimal reject rule for binary classifiers. pages 611–620, 08 2000. doi: 10.1007/3-540-44522-6_63.

Peter D. Turney. Types of cost in inductive concept learning. *CoRR*, cs.LG/0212034, 2002. URL http://arxiv.org/abs/cs.LG/0212034.

Stijn Vanderlooy, Ida G. Sprinkhuizen-Kuyper, Evgueni N. Smirnov, and H. Jaap van den Herik. The roc isometrics approach to construct reliable classifiers. *Intell. Data Anal.*, 13(1):3–37, January 2009. ISSN 1088-467X. URL http://dl.acm.org/citation.cfm?id=1551758.1551760.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

I-Cheng Yeh and Che hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2, Part 1):2473 – 2480, 2009. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2007.12.020. URL http://www.sciencedirect.com/science/article/pii/S0957417407006719.

Erklärung

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer oder der Verfasserin/des Verfassers selbst entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Ort, Datum	Rechtsverbindliche Unterschrift