A Comparison of Two Approaches to Classify with Guaranteed Performance

Stijn Vanderlooy¹ and Ida G. Sprinkhuizen-Kuyper²

¹ MICC-IKAT, Universiteit Maastricht, P.O. Box 616, 6200 MD Maastricht,
The Netherlands

`s.vanderlooy@micc.unimaas.nl`

² NICI, Radboud University Nijmegen, P.O. Box 9104, 6500 HE Nijmegen,
The Netherlands

`i.kuyper@nici.ru.nl`

Abstract. The recently introduced transductive confidence machine approach and the ROC isometrics approach provide a framework to extend classifiers such that their performance can be set by the user prior to classification. In this paper we use the k -nearest neighbour classifier in order to provide an extensive empirical evaluation and comparison of the approaches. From our results we may conclude that the approaches are competing and promising generally applicable machine learning tools.

1 Introduction

In the past decades supervised learning algorithms have been applied to solve various classification tasks with growing success. However, it remains difficult to apply the learned classifiers in domains where incorrect classifications have high costs. Examples of such domains are medical diagnosis and law enforcement.

Recently two approaches have been introduced that extend an existing classifier such that the performance can be set by the user prior to classification. The approaches are called the transductive confidence machine (TCM) approach [1] and the receiver operating characteristic (ROC) isometrics approach [2]. They can construct reliable classifiers since a desired performance can be guaranteed. The key idea of both approaches is to identify instances for which there is uncertainty in the true label. These uncertain instances are inherent to the classification task and they lead to (unaffordable) incorrect classifications. The TCM approach assigns multiple labels to uncertain instances and the ROC isometrics approach leaves uncertain instances unclassified. The approaches are novel with respect to existing reject rules since they are classifier independent and they can guarantee any preset performance. This is not the case when, for example, thresholding the posterior probabilities of a naive Bayes classifier: an upper bound on the error is only guaranteed when the prior is correct [1].

In this paper we use the k -nearest neighbour classifier in combination with the TCM approach and the ROC isometrics approach. In this way we obtain two new classifiers. We use benchmark datasets to verify if the classifiers indeed guarantee a preset performance. We also compare them by analyzing which and

how many instances are considered as uncertain. The best classifier deals with as few uncertain instances as possible in order to guarantee the preset performance.

The remainder of this paper is as follows. Section 2 defines the problem statement that we address. Sections 3 and 4 explain the TCM approach and the ROC isometrics approach, respectively. Section 5 investigates empirically how well both approaches perform and a comparison between them is given. Section 6 provides a discussion. Finally, Section 7 concludes that the approaches are competing and promising generally applicable machine learning tools.

2 Problem Statement

We consider the supervised machine learning setting. The instance space is denoted by $\mathcal{X}$ and the label space by $\mathcal{Y}$. An example is of the form $z = (x, y)$ with $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Training data are considered as a sequence of iid examples:

$$S = (x_1, y_1), \dots, (x_n, y_n) = z_1, \dots, z_n . \quad (1)$$

We desire that the performance of a classifier can be set by the user prior to classification. For example, if the user specifies an accuracy of 95%, then the percentage of incorrect classifications may not exceed 5%. Preset performance and empirical performance are used to denote the user-specified performance and the performance of the classifier, respectively. Thus, empirical performance should be at least preset performance. We also desire that the classifier is efficient in the sense that the number of instances with a reliable and useful classification is high. A classifier is not efficient when it outputs many possible labels for most instances or when it refuses to classify most instances.

3 Transductive Confidence Machine Approach

The TCM approach allows classifiers to assign a set of labels to instances. These sets are called prediction sets. A prediction set contains more than one label if there is uncertainty in the true label of the instance.

Subsection 3.1 explains the construction of prediction sets. Subsection 3.2 shows how to use the k -nearest neighbour classifier (k -NN) as a TCM and Subsection 3.3 shows how to assess the quality of a TCM.

3.1 Algorithm

To construct a prediction set for an unlabeled instance x_{n+1} , TCMs operate in a transductive way [1,3]. Each possible label $y \in \mathcal{Y}$ is tried as a label for instance x_{n+1} . In each try, the example $z_{n+1} = (x_{n+1}, y)$ is formed and added to the training data. Then, each example in the extended sequence:

$$(x_1, y_1), \dots, (x_n, y_n), (x_{n+1}, y) = z_1, \dots, z_{n+1} , \quad (2)$$

is assigned a nonconformity score by means of a nonconformity measure. This measure defines how nonconforming an example is with respect to other available examples.¹ The nonconformity score of example z_i is denoted by α_i .

Since nonconformity scores can be scaled arbitrary, the nonconformity score α_{n+1} is compared to all other α_i in order to know how nonconforming the artificially created example z_{n+1} is in the extended sequence.

Definition 1. *Given a sequence of nonconformity scores $\alpha_1, \dots, \alpha_{n+1}$ with $n \geq 1$, the p-value of label y assigned to an unlabeled instance x_{n+1} is defined as:*

$$p_y = \frac{|\{i = 1, \dots, n+1 : \alpha_i \geq \alpha_{n+1}\}|}{n+1} . \quad (3)$$

If the p-value is close to its lower bound $1/(n+1)$, then example z_{n+1} is very nonconforming. The closer the p-value is to its upper bound 1, the more conforming example z_{n+1} is. Hence, the p-value indicates how likely it is that the tried label is in fact the true label of the unlabeled instance. A TCM outputs the set of labels with p-values above a preset significance level ϵ .

Definition 2. *A transductive confidence machine determined by some nonconformity measure is a function that maps each sequence of examples $z_1, \dots, z_n$ with $n \geq 1$, unlabeled instance x_{n+1} , and significance level $\epsilon \in [0, 1]$ to the prediction set:*

$$I^\epsilon(z_1, \dots, z_n, x_{n+1}) = \{y \in \mathcal{Y} \mid p_y > \epsilon\} . \quad (4)$$

Given a preset significance level ϵ , the performance of a TCM measured by means of accuracy is $100(1 - \epsilon)\%$ [1]. For instance, if the user specifies a significance level of 0.05, then 5% of the computed prediction sets do not contain the true label of the corresponding instances. Thus, the performance is 95%.

3.2 TCM- k NN

Any classifier can be applied as a TCM when the used nonconformity measure identifies an example as nonconforming when its classification is uncertain. This subsection reviews TCM- k NN, a TCM based on k -NN [5].

According to k -NN, an example is nonconforming when it is far from nearest neighbours of the same class and close to nearest neighbours of different classes. So, given example $z_i = (x_i, y_i)$, define an ascending ordered sequence $D_i^{y_i}$ with distances from instance x_i to its nearest neighbours with label y_i . Similarly, let $D_i^{-y_i}$ contain ordered distances from instance x_i to its nearest neighbours with label different from y_i . The nonconformity score is then defined as:

$$\alpha_i = \frac{\sum_{j=1}^k D_{ij}^{y_i}}{\sum_{j=1}^k D_{ij}^{-y_i}} , \quad (5)$$

¹ The next subsection provides a nonconformity measure for the k -nearest neighbour classifier. Measures for other classifiers are found in [1,4] and references therein.

with subscript j representing the j -th element in a sequence [5]. Clearly, the nonconformity score is monotonically increasing when distances to nearest neighbours of the same class increase and/or distances to nearest neighbours of different classes decrease.

3.3 Quality Assessment

The quality of TCMs is assessed by two key statistics. The first statistic is the percentage of prediction sets that contain the true label. This is the empirical performance and its value should be at least the preset performance. The second statistic is efficiency. Efficiency indicates how useful the prediction sets are and it is measured as the percentage of prediction sets with exactly one label. Prediction sets with multiple labels indicate that each of these labels may be correct.²

4 ROC Isometrics Approach

The key idea of the ROC isometrics approach is to leave some instances unclassified if there is uncertainty in the true label of those instances. This is done in such a way that a preset performance is guaranteed.

Subsection 4.1 outlines the approach and Subsection 4.2 shows how it can be used for k -NN. Subsection 4.3 shows how to perform quality assessment. For simplicity, we assume a classification task with a positive and negative class.

4.1 Algorithm

The ROC isometrics approach requires two positive values $l(x|p)$ and $l(x|n)$ that indicate the likelihood that an instance x is positive and negative, respectively. The likelihood values are combined into a score.

Definition 3. *The score of instance x is defined as:*

$$l(x) = \frac{l(x|p)}{l(x|n)} , \quad (6)$$

or ∞ if $l(x|n) = 0$.

Scores are used to rank instances from most likely positive to most likely negative [6]. An unlabeled instance is classified as positive when its score is at least the value of some threshold.

Definition 4. *An ROC curve is a plot with false positive rate (fpr) on the horizontal axis and true positive rate (tpr) on the vertical axis. It shows fpr and tpr values for each possible threshold on scores [7].*

² Prediction sets can also be empty. However, most empty prediction sets occur when the preset significance level results in a TCM with lower performance than is achieved by the conventional classifier [1,4]. This is clearly not an interesting situation.

Theorem 1. *For any point on and below the convex hull of an ROC curve (ROCCH), a classifier can be constructed by thresholding the scores in such a way that it achieves the fpr and tpr values represented by that point [7].*

In the ROC isometrics approach, the user has to preset a desired performance for each class. Positive class performance is defined as the fraction of positive classifications that are correct. Negative class performance is defined analogously. For each preset performance, an ROC isometric is constructed.

Definition 5. *ROC isometrics are curves that connect points with the same performance in the (fpr, tpr) plane [8].*

The intersection point of a positive (negative) class isometric and the ROCCH represents a classifier with the positive (negative) class performance as preset by the user. The intersection point of the two isometrics themselves represents the classifier that guarantees the preset performance on both classes (henceforth simply called performance). We distinguish three cases as shown in Fig. 1.

- **Case 1:** the isometrics intersect on the ROCCH.
The classifier that guarantees the preset performance lies on the ROCCH. From Theorem 1 it follows that the classifier can be constructed.
- **Case 2:** the isometrics intersect below the ROCCH.
Theorem 1 also applies in this case. However, classifiers corresponding to points on the ROCCH between the intersection points of the isometrics and the ROCCH have higher performance.³
- **Case 3:** the isometrics intersect above the ROCCH.
Theorem 1 does not apply in this case. The proposed solution is to identify two thresholds $a > b$ that correspond with the intersection points of the positive class isometric and the ROCCH, and the negative class isometric and the ROCCH, respectively. A new instance x_{n+1} is classified as positive if $l(x_{n+1}) \geq a$ and as negative if $l(x_{n+1}) \leq b$. Otherwise, the instance is left unclassified since its classification is uncertain. It has been shown that the resulting classifier guarantees the preset performance (an unclassified instance is not counted as an error) [2].

4.2 ROC- k NN

Any classifier can be used to construct an ROC curve. Some classifiers such as naive Bayes and neural networks naturally provide likelihood values. For other classifiers, such as k -NN, a post-processing technique is needed.

We computed the likelihood value of instance x_i according to k -NN by computing the inverse of the nonconformity score (5). Our result of using k -NN in the ROC isometrics approach is denoted by ROC- k NN.

³ This case is analogous to the case of choosing a significance level in the TCM approach such that empty prediction sets occur (i.e., the preset performance is lower than can be achieved by the conventional classifier).

Fig. 1. Location of the intersection point of a positive class isometric (—) and a negative class isometric (---): (a) Case 1, (b) Case 2, and (c) Case 3

4.3 Quality Assessment

Analogously to the TCM approach, quality assessment of the ROC isometrics approach is performed by reporting empirical performance (accuracy) and efficiency. Efficiency is now measured by the percentage of instances for which a label is predicted.

5 Experiments

This section provides the experimental results. We performed experiments with TCM- k NN and ROC- k NN on a number of benchmark datasets. Subsection 5.1 describes these datasets and Subsection 5.2 outlines the experimental setup. Subsection 5.3 evaluates and compares the two classifiers in terms of empirical performance and efficiency.

5.1 Benchmark Datasets

We tested TCM- k NN and ROC- k NN on six well-known binary datasets from the UCI benchmark repository [9] and four binary datasets from a recent machine learning competition [10]. The datasets vary greatly in size and in class ratio. The classes are denoted by positive class (p) and negative class (n). As a preprocessing step, all instances with missing feature values are removed as well as duplicate instances. Features are standardized to have zero mean and unit variance. Finally, linear discriminant analysis is used to project the data into a linear one-dimensional space. This allows for a post-hoc visualization (see Section 5.3). Table 1 summarizes the main characteristics of the datasets.

5.2 Experimental Setup

Nearest neighbours are found with Euclidean distances. The number of nearest neighbours is restricted to $k = 1, 2, \dots, 10$ and chosen such that the average accuracy of k -NN on the test folds is maximized (using 10-fold cross validation).

Table 1. Benchmark datasets: name, number of examples, minority class, percentage of examples in the minority class, and the accuracy of k -NN (10-fold cross validation). The first six datasets are from the UCI repository, the last four are from the competition.

name	size	min. class	% min. class	accuracy
heart statlog	270	p	44.44	84.81
house votes	342	p	34.21	94.13
ionosphere	350	n	35.71	90.57
monks3	432	n	48.15	78.70
sonar	208	p	46.63	89.90
spect	219	n	12.79	88.12
ada	4560	p	24.81	83.55
gina	3460	p	49.16	93.31
hiva	4220	p	3.52	97.22
sylva	14390	p	6.15	97.57

Once the value of k is chosen for each dataset, the classifiers TCM- k NN and ROC- k NN are applied using 10-fold cross validation. For TCM- k NN, each test instance is used in combination with the training fold to compute its prediction set. For ROC- k NN, each training fold is used to construct an ROCCH in order to find the threshold(s). We report on the average performance (accuracy) and average efficiency over the test folds. To compare both classifiers on efficiency, we say that TCM- k NN leaves an instance unclassified when the corresponding prediction set contains multiple labels. To construct ROC isometrics, we set both the positive class performance and the negative class performance equal to the preset performance. In this way, a preset performance (accuracy) is obtained [2]. We consider five preset performances that we believe to be of interest in many classification tasks: 95%, 96%, 97%, 98%, and 99%.⁴

5.3 Quality Assessment of TCM- k NN and ROC- k NN

In this section we report on the experimental results. Results on the `hiva` and `sylva` datasets are omitted for preset performances below 98% since the conventional classifier has higher performance. No results of ROC- k NN on the `spect` dataset are obtained since the negative class isometric did not intersect the (badly structured) ROCCH.

Table 2 shows that the empirical performance equals the preset performance up to statistical fluctuations, even for the small datasets. These results verify that the performance of both classifiers can be preset. Since some datasets have a highly unbalanced class distribution, it is desired that the empirical performances on each class are approximately equal. Table 3 shows the differences between the positive class performances and the negative class performances. A positive value indicates that the positive class performance is higher than the

⁴ We do not consider 100% preset performance since a TCM will leave all instances unclassified. A comparison with the ROC isometrics approach is then inappropriate.

negative class performance. The differences are large for TCM- k NN when it is applied on datasets with an unbalanced class distribution. The sign of these differences shows that the classifier uses easy-to-classify instances from the majority class to mask bad performance on the minority class, except for **ionosphere** and **monks3**. The large differences for **spect** are explained by a positive class performance of 100% for all preset performances. As expected, the differences for ROC- k NN show that the preset performance is guaranteed for both classes, with the exception of **ionosphere**, **monks3**, and **ada**. For these datasets, the classifier seems to suffer more from statistical fluctuations. In [2] it is proved that the ROC isometrics approach guarantees a preset performance on each class.

Table 4 shows the efficiency (i.e., the percentage of classified instances) of TCM- k NN and ROC- k NN. In general, efficiency declines exponentially when the preset performance is increased. There is no clear relation between dataset characteristics and efficiency, e.g., datasets with a highly unbalanced class distribution such as **hiva** and **sylva** can still have relatively few unclassified instances. In addition, neither of the two classifiers can be claimed as the most efficient. Noteworthy is the bad efficiency of ROC- k NN on **ionosphere**, **ada**, and **hiva** compared to that of TCM- k NN. This is due to a large number of positive instances and negative instances for which the likelihood values did not result in good scores to discriminate both classes. However, ROC- k NN is the most efficient on the majority of the remaining datasets.

An unclassified instance should be an instance for which there is uncertainty in the true label. We verified this by checking visually if TCM- k NN and ROC- k NN leave instances unclassified that lie in or close to the overlap of the class data histograms. Figure 2 gives an example. Since the number of unclassified instances is limited, we expect that many instances left unclassified by TCM- k NN and ROC- k NN are identical. Table 5 (left part) verifies a large percentage of identical unclassified instances that, in general, increases when the preset performance increases.⁵ However, focusing on identical unclassified instances underestimates the resemblance of the two classifiers since unclassified instances very close to each other should also be considered as identical. Therefore, Table 5 (right part) shows the percentage of unclassified instances that are approximately identical, i.e., unclassified instances of both classifiers that are among the ten nearest neighbours of each other. The results clearly show that the classifiers are similar in terms of identifying instances with uncertainty in the true label.

6 Discussion

Our experimental results show that TCM- k NN and ROC- k NN are competing classifiers in terms of guaranteed performance and efficiency. This section provides a discussion on these results by elaborating on some noteworthy differences between the TCM approach and the ROC isometrics approach.

⁵ Reported values are computed as the fraction of the number of equal unclassified instances and the minimum number of unclassified instances of the two classifiers. The fraction is then converted to a percentage.

Table 2. Empirical performances of TCM-*k*NN (left part) and ROC-*k*NN (right part): the preset performances are guaranteed up to statistical fluctuations, even for small datasets

dataset	95%	96%	97%	98%	99%	95%	96%	97%	98%	99%
heart	95.2	96.0	96.7	98.5	99.3	94.5	95.6	97.0	98.0	98.8
house votes	95.3	95.9	97.1	97.9	99.1	95.8	96.1	97.2	97.7	98.7
ionosphere	95.4	96.3	97.1	98.3	98.7	95.1	95.0	96.1	97.5	98.6
monks3	95.1	95.8	97.4	98.1	99.3	95.6	96.0	96.9	97.7	99.0
sonar	95.5	96.5	97.5	98.5	99.5	94.8	95.9	97.0	98.1	98.8
spect	95.2	96.2	97.6	98.6	99.5	-	-	-	-	-
ada	95.0	96.0	97.0	98.0	99.0	96.0	96.9	97.8	98.5	99.3
gina	95.0	96.0	97.0	98.1	99.1	95.0	96.0	97.0	98.0	99.0
hiva	-	-	-	98.0	99.0	-	-	-	98.0	99.0
sylva	-	-	-	98.0	99.0	-	-	-	98.0	99.0

Table 3. Differences between the empirical performances on each class of TCM-*k*NN (left part) and ROC-*k*NN (right part): TCM-*k*NN gives empirical performances that, in general, are far less balanced over the classes than is the case with ROC-*k*NN

dataset	95%	96%	97%	98%	99%	95%	96%	97%	98%	99%
heart	1.0	1.0	-0.4	-0.7	-1.9	0.2	1.8	-2.5	-2.5	-2.0
house votes	-7.9	-6.2	-5.3	-4.2	-2.4	-2.1	-1.7	-1.5	-0.1	-2.1
ionosphere	-5.7	-4.6	-4.3	-2.8	-1.9	-1.7	-1.9	-7.4	-10.0	-8.3
monks3	7.4	7.0	5.2	3.7	1.6	5.0	7.8	6.9	1.3	1.2
sonar	1.7	0.0	-1.0	-0.8	-1.0	-0.2	-0.1	-2.2	1.1	-1.3
spect	32.5	26.2	19.5	7.0	3.33	-	-	-	-	-
ada	-12.1	-9.8	-7.4	-5.4	-2.6	-5.0	-6.5	-4.5	-5.0	-6.9
gina	0.5	0.6	0.1	-0.5	-0.3	0.0	0.0	-0.2	-0.5	-0.6
hiva	-	-	-	-33.7	-19.7	-	-	-	-0.6	-1.3
sylva	-	-	-	-24.0	-13.3	-	-	-	-0.2	0.1

The TCM approach extends a classifier such that any preset performance (accuracy) can be guaranteed when a consistent nonconformity measure is used [1,4,5]. In contrast, the highest performance that can be achieved using the ROC isometrics approach depends on the ROCCH. Any preset performance can be guaranteed when the first line segment is vertical and the last line segment is horizontal. This is the case when the highest score and the lowest score are assigned exclusively to positive instances and negative instances, respectively. A few outliers or noise in the data can therefore cause that some preset performances cannot be guaranteed. In addition, randomization of two thresholds on scores is needed when an isometric does not intersect the ROCCH in an end-point of two adjacent line segments [7]. For small datasets this can result in a deviation from preset performance, although in our experiments the deviations are negligible, as seen in Table 6. Finally, limited amount of data can yield an ROC curve that is a bad estimate of the true curve.

Table 4. Efficiency of TCM-*k*NN (left part) and ROC-*k*NN (right part): no classifier is clearly more efficient than the other classifier

dataset	95%	96%	97%	98%	99%	95%	96%	97%	98%	99%
heart	65.7	55.9	52.0	46.0	31.7	55.6	54.0	47.0	40.0	35.0
house votes	99.0	97.6	95.8	93.7	79.0	99.4	99.4	97.6	95.9	87.9
ionosphere	83.9	81.2	78.3	72.3	50.5	84.0	78.0	52.0	35.7	30.0
monks3	71.2	69.6	33.2	31.3	28.9	57.2	53.3	52.8	51.2	44.2
sonar	89.2	85.9	78.6	73.6	64.4	90.0	84.5	79.5	77.5	72.5
spect	43.0	38.4	34.7	27.4	18.7	-	-	-	-	-
ada	57.3	53.6	48.9	41.6	28.6	42.6	37.8	31.9	25.2	19.6
gina	96.5	93.8	90.3	84.7	74.0	96.5	93.8	89.5	83.8	67.2
hiva	-	-	-	98.2	92.0	-	-	-	80.0	73.6
sylva	-	-	-	97.9	95.0	-	-	-	99.5	96.6

Fig. 2. Unclassified instances of TCM-*k*NN (□) and ROC-*k*NN (O) for `heart statlog` dataset with a preset performance of 95%: the dark region in the middle of the class histograms shows the overlap of the histograms (the region of uncertainty).

On the other hand, the ROC isometrics approach has a significant computational advantage since a classifier is only trained once and subsequently used to construct an ROCCH. In addition, the approach is able to incorporate different costs of incorrect classifications, e.g., one can specify that classifying a negative instance incorrectly is more severe than classifying a positive instance incorrectly. The ability to incorporate costs is important for two reasons: (1) most application domains have a non-uniform cost distribution and (2) the cost distribution often changes in time. Costs are incorporated via the isometrics since the (fpr, tpr) plane is independent of the cost distribution. Finally, isometrics can be constructed for a variety of performance metrics such as the m -estimate and the F -measure [2].

The TCM approach can be applied without modification to multi-class classification problems. This is not the case for the ROC isometrics approach: given C classes, the search space has dimension $C^2 - C$ and isometrics are not yet

Table 5. Comparison of unclassified instances: identical unclassified (left part) and approximately identical unclassified (right part)

dataset	95%	96%	97%	98%	99%	95%	96%	97%	98%	99%
heart	84.6	84.8	93.0	92.4	95.4	97.8	93.2	100.0	100.0	98.9
house votes	0.0	33.3	71.4	81.3	73.9	100.0	100.0	100.0	100.0	82.6
ionosphere	45.5	55.6	65.8	69.9	80.0	64.5	66.7	75.3	82.8	85.9
monks3	86.2	87.7	94.2	94.9	90.2	97.6	95.4	96.6	97.7	91.0
sonar	100.0	92.3	90.2	95.6	100.0	100.0	96.2	100.0	100.0	100.0
spect	-	-	-	-	-	-	-	-	-	-
ada	100.0	100.0	100.0	100.0	98.7	100.0	100.0	100.0	100.0	100.0
gina	82.5	89.5	99.4	98.1	93.5	84.2	95.2	100.0	100.0	100.0
hiva	-	-	-	86.8	100.0	-	-	-	100.0	100.0
sylva	-	-	-	79.4	94.4	-	-	-	81.0	96.6

Table 6. Standard deviations of the empirical performance by applying ROC- k NN ten times: randomization of thresholds has minor influence, even for small datasets

dataset	95%	96%	97%	98%	99%
heart	0.77	0.44	1.08	0.95	1.03
house votes	0.44	0.36	0.26	0.51	0.23
ionosphere	0.33	0.33	0.42	1.35	0.40
monks3	0.76	0.62	0.65	0.44	0.44
sonar	0.34	0.65	0.45	0.60	0.43
spect	-	-	-	-	-
ada	0.08	0.12	0.25	0.13	0.17
gina	0.07	0.06	0.03	0.04	0.06
hiva	-	-	-	0.09	0.02
sylva	-	-	-	0.02	0.01

investigated in this space. (See [6] for an overview of approaches to multi-class ROC analysis.) We are currently extending the ROC isometrics approach to multi-class classification problems.

7 Conclusions

In this paper we used the k -nearest neighbour classifier in combination with the TCM approach and the ROC isometrics approach. The two resulting classifiers are applied on ten benchmark datasets in order to provide an extensive empirical evaluation and comparison in terms of performance and efficiency.

We review our contributions and formulate three conclusions. First, we verified that ROC- k NN guarantees a preset performance. This is also the case for TCM- k NN, as pointed out by earlier results [1,4]. Experiments with a naive Bayes classifier gave similar results. We may conclude that, dependent on the classification task, the user can preset the performance such that the number of incorrect classifications that may still occur is acceptable. Second, the

approaches can identify instances for which there is uncertainty in the true label. These instances are difficult to classify and analyzing them can result in a better understanding of the problem at hand. Therefore, we may conclude that the approaches provide valuable feedback to the user. Third, we discussed the advantages and disadvantages of each approach. We conclude that the ROC isometrics approach is preferred in the following four situations: (1) fast processing of instances, (2) a balanced performance over the classes, (3) non-uniform cost distribution, and (4) choice of performance metric. The TCM approach is preferred in the following two situations: (1) low performance of a conventional classifier (badly structured ROCCH), and (2) limited amount of data. Clearly, the approaches are generally applicable and promising machine learning tools that should find their way into practice.

Acknowledgments

We thank the reviewers for useful suggestions. The first author is supported by the Dutch Organization for Scientific Research (NWO), grant nr: 634.000.435.

References

1. Vovk, V., Gammerman, A., Shafer, G.: *Algorithmic Learning in a Random World*. Springer, New York (2005)
2. Vanderlooy, S., Sprinkhuizen-Kuyper, I., Smirnov, E.: An analysis of reliable classifiers through ROC isometrics. In: Lachiche, N., Ferri, C., Macskassy, S., ICML (eds.) 2006 Workshop on ROC Analysis (ROCML 2006), Pittsburgh, USA, pp. 55–62 (2006)
3. Gammerman, A., Vovk, V.: Prediction algorithms and confidence measures based on algorithmic randomness theory. *Theoretical Computer Science* 287, 209–217 (2002)
4. Vanderlooy, S., van der Maaten, L., Sprinkhuizen-Kuyper, I.: Off-line learning with transductive confidence machines: an empirical evaluation. Technical Report 07-03, Universiteit Maastricht, Maastricht, The Netherlands (2007)
5. Proedrou, K., Nouretdinov, I., Vovk, V., Gammerman, A.: Transductive confidence machines for pattern recognition. Technical Report 01-02, Royal Holloway University of London, London, UK (2001)
6. Lachiche, N., Flach, P.: Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves. In: Fawcett, T., Mishra, N. (eds.) 20th International Conference on Machine Learning (ICML 2003), pp. 416–423. AAAI Press, Washington, DC, USA (2003)
7. Provost, F., Fawcett, T.: Robust classification for imprecise environments. *Machine Learning* 42, 203–231 (2001)
8. Flach, P.: The geometry of ROC space: Understanding machine learning metrics through ROC isometrics. In: Fawcett, T., Mishra, N. (eds.) 20th International Conference on Machine Learning (ICML 2003), pp. 194–201. AAAI Press, Washington, DC, USA (2003)
9. Newman, D., Hettich, S., Blake, C., Merz, C.: *UCI repository of machine learning databases* (1998)
10. Guyon, I.: Data representation discovery workshop of the 20th international joint conference on neural networks (IJCNN 2007) (2007)