Meta-conformity approach to reliable classification

E.N. Smirnov\textsuperscript{a}, G.I. Nalbantov\textsuperscript{a} and A.M. Kaptein\textsuperscript{b}

\textsuperscript{a}Department of Knowledge Engineering, Faculty of Humanities and Sciences, Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands
E-mail: \{smirnov.g.nalbantov\}@micc.unimaas.nl

\textsuperscript{b}Archives and Information Studies, University of Amsterdam, 1012 XT Amsterdam, The Netherlands
E-mail: a.m.kaptein@uva.nl

Abstract. The conformity framework has recently been proposed for the task of reliable classification. Given a classifier \( B \), the framework allows to obtain \( p \)-values of the classifications assigned to individual instances. However, applying the framework is a difficult problem: we need to construct an instance non-conformity function for the classifier \( B \). To avoid constructing such a function we propose a meta-conformity approach.\textsuperscript{1} If a conformity-based classifier \( M \) is available, the approach is to train \( M \) as a meta classifier that predicts the correctness of each classification of the classifier \( B \). In this way the classification \( p \)-values of the classifier \( B \) are represented by the classification \( p \)-values of the classifier \( M \).

The meta-conformity approach can be used for constructing classifiers with predefined generalization performance. Experiments show that the approach results in classifiers that can outperform existing conformity-based classifiers.

Keywords: Reliable classification, conformity framework, meta classification

1. Introduction

Classifiers applied in critical domains need to determine whether classifications they assign to individual instances are indeed reliable [19,24]. To decide whether an instance classification is reliable we consider two tasks (cf. [2,4,5,11,19,23,24,26]):

(T1) to obtain a reliability value for that classification; and
(T2) to obtain a threshold on the classification-reliability values.

If the reliability value for the instance classification is greater than the threshold, the classification is considered reliable. Otherwise, it is unreliable and the instance is left unclassified.

The task T1 of obtaining classification-reliability values has extensively been studied in [2,4,5,11,19,21,23,24,26]. One of the most successful approaches to the task is the conformity framework proposed in [19,24]. The framework allows to compute a \( p \)-value for each classification a classifier assigns to an instance if the data are generated by the same unknown exchangeability probability distribution. However, to apply the framework for a classifier we need to compute instance non-conformity values. To compute these values we need to construct a non-conformity function specific for that classifier.

\textsuperscript{1}This paper is an extended version of [20].
Constructing non-conformity functions is a difficult problem, since there is no approach on how to construct these functions in general [19,24]. Therefore, the applicability of the conformity framework is still restricted.

In order to apply the conformity framework for any classifier \( B \) we propose in this paper a meta-conformity approach inspired by [3,12,14]. If a conformity-based classifier \( M \) is available, the approach is to train \( M \) as a meta classifier that predicts the correctness of each classification of the classifier \( B \). In this way the classification \( p \)-values of the classifier \( B \) are represented by the classification \( p \)-values of the classifier \( M \). Thus, the conformity framework becomes applicable for the classifier \( B \).

The task T2 of obtaining thresholds on classification-reliability values has been studied in [8,9,16,23] using concepts from the ROC analysis [7]. To solve the task for the meta-conformity approach we propose in this paper a ROC-based procedure for classifier construction. The procedure is based on our results described in [23]. It allows constructing classifiers for reliable classification with predefined generalization performance.

The paper is organized as follows. The necessary background information is provided in Sections 2, 3, and 4. Section 2 introduces the reliable classification task. The performance metrics for classifier evaluation are presented in Section 3. Section 4 describes briefly the conformity framework. The next four Sections introduce the meta-conformity approach in detail. Section 5 describes the main components of the approach. Our ROC-based procedure for classifier construction is presented in Section 6. Section 7 describes experiments. A comparison with relevant work is provided in Section 8. Finally, Section 9 concludes the paper.

2. Reliable classification task

The task of reliable classification is a sub-task of the classification task [21]. Let \( X \) be an instance space and \( Y \) be a class set. The training data \( D \) are represented by a multiset \(^2\) \( \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \) of \( n \) instances \( (x_i, y_i) \in X \times Y \) drawn from the same unknown probability distribution. Given a space \( H \) of classifiers \( h : X \rightarrow Y \) and the training data \( D \), the reliable classification task is to find classifier \( h \in H \) that correctly classifies future, unseen instances \( x \in X \); when correct classification is not possible for an instance \( x \in X \), the classification process outputs symbol “?” and the instance \( x \) is left unclassified.

3. Metrics for classifier performance

To decide whether to employ a classifier, we need to evaluate its classification performance. Below we consider the main metrics for classifier evaluation. For the sake of simplicity the metrics are given for binary classifiers.

Given a binary classifier \( h \in H \) and a multiset of test instances, we can construct a confusion matrix (see the first two rows of the matrix in Fig. 1). The matrix gives the counts of the test instances depending on their true class and hypothesized class provided by the classifier \( h \). Counts of true positive instances, false positive instances, true negative instances, and false negative instances are denoted by \( TP \), \( FP \), \( TN \), and \( FN \), respectively.

\(^2\)The training data are viewed as a multiset, since the instances can be repeated. The notation of multisets employed in this paper is that introduced in [22].
In case of reliable classification a classifier $h$ can abstain from uncertain classifications (indicated by the symbol "?"). Therefore, the confusion matrix is extended by an additional row of counts of unclassified positive instances $UP$ and unclassified negative instances $UN$.

The counts from a confusion matrix are used for deriving useful metrics characterizing the performances of classifiers for reliable classification. We provide first the most basic metrics: true positive rate ($TPr$), false positive rate ($FPr$), true negative rate ($TNr$), and false negative rate ($FNr$) defined as follows:

$$TPr = \frac{TP}{TP + FN}$$
$$FPr = \frac{FP}{FP + TN}$$
$$FNr = \frac{FN}{TP + FN}$$
$$TNr = \frac{TN}{FP + TN}$$

Additional metrics that are used throughout this paper are accuracy rate $A$ and precision rate $P$ for the positive class. They are defined as follows:

$$A = \frac{TP + TN}{TP + FP + FN + TN}$$
$$P = \frac{TP}{TP + FP}$$

We note that the performance metrics of a classifier $h \in H$ for reliable classification given above are defined only on the instances that can be classified by $h$. To characterize the instances that are classified and the instances that are not classified we employ two mutually complimentary metrics: rejection rate and coverage rate. The rejection rate is the proportion of instances not classified by $h$. The coverage rate equals one minus the rejection rate. We expect that increasing the number of the rejected instances implies lower number of misclassifications. Therefore, in this paper we employ the rejection rate $R$ defined as follows:

$$R = \frac{UP + UN}{TP + FP + FN + TN + UP + UN}$$
The conformal framework has been proposed for obtaining a \( p \)-value for each class a classifier can assign to an instance being classified \([19,24]\). The framework has been proven to be valid when the data are drawn from the same unknown distribution under the exchangeability assumption. The later holds when different orderings on any multiset of instances are equally likely.

Applying the conformal framework is a two-stage process. Given a classifier \( h \in H \), we first construct a non-conformity function \( A \) capable of measuring how unusual an instance looks relatively to the other instances in the data. Then, we apply the conformal algorithm that employs the conformity function to compute the classification \( p \)-values.

Formally, a non-conformity function \( A \) is a function that maps a multiset \( D \) of instances defined in \( X \times Y \) and an instance \( (x, y) \in X \times Y \) into a value \( \alpha \in [0, R \cup \{\infty\}] \) indicating how unusual the instance \((x, y)\) for the instances in \( D \). To provide an example, let us consider the non-conformity function \( A \) for the nearest neighbor classifiers proposed in \([17]\). Given an instance \((x, y) \in X \times Y \), the non-conformity function \( A \) returns for \((x, y)\) a non-conformity value \( \alpha \) equal to \( d_{k}^{-y} \) where \( d_{k}^{-y} \) is the sum of distances between \( x \) and \( k \) nearest neighbors of \( x \) in \( D \) that belong to class \( y \), and \( d_{k}^{-y} \) is the sum of distances between \( x \) and \( k \) nearest neighbors of \( x \) in \( D \) that do not belong to class \( y \).

The conformal algorithm is presented in Fig. 2. Given training data \( D \) of \( n \) instances, an instance \( x_{n+1} \in X \) to be classified, the non-conformity function \( A \) for a classifier \( h \in H \), and a class \( y \in Y \), the algorithm computes \( p \)-value \( p_{y} \) for class \( y \) when it is assigned to \( x_{n+1} \). To compute the \( p \)-value \( p_{y} \), the algorithm first adds the instance \((x_{n+1}, y)\) to the training data \( D \). Then, it computes the non-conformity score \( \alpha \) for each instance \((x_{i}, y_{i}) \in D \) using the non-conformity function \( A \). The non-conformity scores are used for computing the \( p \)-value \( p_{y} \) of the class \( y \) for the instance \( x_{n+1} \). More precisely, \( p_{y} \) is computed as the proportion of the instances in the training data \( D \) of which the non-conformity scores \( \alpha \) are greater than or equal to that of the instance \((x_{n+1}, y)\).

To provide an example we note that the conformal algorithm has been applied for the nearest neighbor classifiers in \([17]\). The algorithm has employed the nearest neighbor non-conformity function presented above. The resulting classifier is called the conformity-based nearest neighbor classifier (CNN) and it is employed in the experimental part of this paper.

The main problems of the conformity framework come from the non-conformity functions. Although these functions have been proposed for some basic types of classifiers (e.g., the nearest neighbor classifiers \([17]\), support vector machines \([18]\)), there are no formal requirements for non-conformity
functions and, thus, there is no methodology about how to design these functions in general. Moreover, if we do not have access to the internal structure of classifiers (e.g., the classifier is a human expert), we cannot design non-conformity functions at all. Therefore, we may conclude that the applicability of the conformity framework is restricted.

5. Meta-conformity approach

We propose the meta-conformity approach to allow the conformity framework to be applied to any type of classifiers so that we do not have to construct a non-conformity function. Assume that we have a classification problem that requires computing $p$-values for instance classifications but our best classifier $B$ is not capable of providing such values. Then, if we have any conformity-based classifier $M$, the approach is to train $M$ as a meta classifier that predicts the correctness of each instance classification of $B$. In this way the $p$-values of the instance classifications of the base classifier $B$ can be represented by the $p$-values of the meta classifications of the meta classifier $M$. Thus, the conformity framework becomes applicable for the classifier $B$.

Below in the next three Subsections we present the main components of the meta-conformity approach.

5.1. Meta classifiers

Assume that we have a base classifier $B$ constructed on the training data $D$. From correct and incorrect classifications of the base classifier $B$ on some test data we can train a meta classifier $M$ [3,12]. The meta classifier $M$ allows to characterize the reliability (correctness) of each classification of the base classifier $B$ (see Figs 3 and 5).

For the meta-conformity approach the meta classifier $M$ is trained on meta data $D'$ generated using the original training data $D$ as follows. The meta data $D'$ are defined in a meta instance space $X'$ given a meta class set $Y'$. The meta instance space $X'$ coincides with the instance space $X$. The meta class set $Y'$ consists of two meta classes: “positive meta class” denoting reliable classification and “negative meta class” denoting unreliable classification. The meta data $D'$ are formed using an internal $k$-fold cross-validation process similarly to [6]. This process splits randomly the training data $D$ into $k$ equally-sized folds $F_i$ ($i \in [1,k]$) (see Fig. 4). Then, for each $j \in [1,k]$ we combine all the folds $F_i$ with $i \neq j$ into
Base Classifier $B$

$F_1$ $F_{j-1}$ $F_j$ $F_{j+1}$ $F_k$

Fig. 4. Meta data generation.

Instance $x$

Base Classifier

$y$

Meta Classifier

$y$ is reliable?

$p_p$ and $p_n$

Fig. 5. The classification rule of the combined classifier $B:M$: an instance $x \in X$ receives a classification $y \in Y$ if and only if the meta classifier $M$ decides that this classification is reliable. The rule is used for approximating the $p$-values of the classifications of the base classifier $B$ by the $p$-values of the meta classifications of the meta classifier $M$.

data $D_j$. The base classifier $B$ is trained on $D_j$. When $B$ is tested against the instances in the fold $F_j$, we generate the meta instances corresponding to the instances of fold $F_j$ as follows. If a labeled instance $(x_i, y_i) \in F_j$ is classified correctly, i.e., the classification of $x_i$ provided by classifier $B$ equals $y_i$, then we generate a meta instance $(x_i, y'_i)$ of which the label $y'_i \in Y'$ equals “positive meta class”. If a labeled instance $(x_i, y_i) \in F_j$ is classified incorrectly, i.e., the classification of $x_i$ provided by classifier $B$ does not equal $y_i$, then we generate a meta instance $(x_i, y'_i)$ of which the label $y'_i \in Y'$ equals “negative meta class”. In both cases the meta instance $(x_i, y'_i)$ is added to the meta fold $F'_j$. After the formation of all the meta folds $F'_j$ ($j \in [1, k]$), the meta data $D'$ are formed as the union of these folds.

Once the meta data $D'$ have been formed, the meta classifier $M$ is trained on $D'$. In this way $M$ is capable of predicting whether instance classifications of the base classifier $B$ are reliable (correct). One important property of the meta data $D'$ is that they are not linearly separable in general, even if the training data $D$ are linearly separable and the base classifier $B$ is linear [12]. Thus, it is desirable that a non-linear classifier is chosen for the meta classifier $M$. 

5.2. Combined classifiers

Classifiers $B$ and $M$ form one combined classifier denoted by $B:M$. The classification rule of the combined classifier $B:M$ is to assign a class $y \in Y$ predicted by $B$ to an instance $x \in X$ if $M$ decides that this classification is reliable (i.e., $M$ assigns “positive meta class” to $x$); otherwise, $x$ is unclassified (see Fig. 5).

According to this classification rule, the combined classifier $B:M$ may not classify all the instances. Therefore, we have to characterize the rejection rate of $B:M$. By Theorem 1 given below the rejection rate $R_{B:M}$ of the combined classifier $B:M$ equals the proportion of the instances for which the meta classifier $M$ assigns the “negative meta class”.

**Theorem 1.** Given a base classifier $B$, a meta classifier $M$, and a combined classifier $B:M$, the rejection rate $R_{B:M}$ of the classifier $B:M$ equals:

$$
\frac{TN_M + FN_M}{TP_M + FP_M + FN_M + TN_M}.
$$

**Proof.** The proof follows from the confusion matrix of the meta classifier $M$. □

By Theorem 2 given below the accuracy rate $A_{B:M}$ of the combined classifier $B:M$ on classified instances equals the precision rate $P_M$ of the meta classifier $M$ for the “positive meta class”.

**Theorem 2.** Given a base classifier $B$, a meta classifier $M$, and a combined classifier $B:M$, the accuracy rate $A_{B:M}$ of the classifier $B:M$ equals the precision rate $P_M$ of the classifier $M$.

**Proof.** From the construction of the combined classifier $B:M$ it follows that:

$$
TP_{B:M} + TN_{B:M} = (TP_B + TN_B) \frac{TP_M}{TP_M + FN_M} \quad (4)
$$

$$
FP_{B:M} + FN_{B:M} = (FP_B + FN_B) \frac{FP_M}{FP_M + TN_M} \quad (5)
$$

But, from the confusion matrices of the base classifier $B$ and meta classifier $M$ we have:

$$
TP_B + TN_B = TP_M + FN_M \quad (6)
$$

$$
FP_B + FN_B = FP_M + TN_M \quad (7)
$$

If we substitute $TP_B + TN_B$ in formula (4) by the right hand side of formula (6) and $FP_B + FN_B$ in formula (5) by the right hand side of formula (7), we receive the following two equations:

$$
TP_{B:M} + TN_{B:M} = TP_M \quad (8)
$$

$$
FP_{B:M} + FN_{B:M} = FP_M \quad (9)
$$

*The performance metrics of the classifiers $B$, $M$, and $B:M$ will be given with subscripts $B$, $M$, and $B:M$, respectively.*
We substitute $TP_{B:M} + TN_{B:M}$ and $FP_{B:M} + FN_{B:M}$ according to formulas (8) and (9) in the equation of the accuracy rate $A_{B:M}$ (see formula (1)). Thus,

$$A_{B:M} = \frac{TP_{B:M} + TN_{B:M}}{TP_{B:M} + TN_{B:M} + FP_{B:M} + FN_{B:M}}$$

But, formula (11) is the precision rate $P_M$ of the meta classifier $M$ for the “positive meta class” (see formula (2)). Thus, we conclude that:

$$A_{B:M} = P_M.$$ 

Theorem 2 shows that the accuracy of the base classifier $B$ does not influence directly the accuracy of the combined classifier $B:M$. Since the accuracy rate $A_{B:M}$ of the combined classifier $B:M$ equals the precision rate $P_M$ of the meta classifier $M$, to maximize $A_{B:M}$ we have to tune the meta classifier $M$ so that $P_M$ is maximized.

5.3. Approximation of $p$-values of the classifications of the combined classifiers

Given a base classifier $B$ and a meta classifier $M$, we need to approximate $p$-values of classifications of the combined classifier $B:M$ when:

(S1) the base classifier $B$ is not capable of outputting $p$-values for instance classifications; i.e., $B$ is not based on the conformity framework (e.g., version spaces, random forest, decision rules, naive Bayes classifiers, bagging, stacking, etc. [10,15]).

In this context, let us assume that:

(S2) the meta classifier $M$ is capable of outputting $p$-values for the “positive meta class” and “negative meta class” for any meta instance; i.e., $M$ is based on the conformity framework (e.g., conformity-based nearest-neighbor classifiers [17] and conformity-based support vector machines [18]).

In the setting provided by (S1) and (S2) let us classify an instance $x \in X$ using the combined classifier $B:M$. According to the classification rule of the combined classifier $B:M$, first the base classifier $B$ classifies the instance $x$ and assigns some class $y \in Y$ to $x$. Then, the instance $x$ is considered on the meta level and the meta classifier $M$ is applied on $x$. The meta classifier $M$ computes a $p$-value $p_p$ for the “positive meta class” and a $p$-value $p_n$ for “negative meta class”. In this context we note that the “positive meta class” denotes that the classification $y$ is correct and that the “negative meta class” denotes that the classification $y$ is not correct. Thus, we arrive at our meta-conformity assumptions:

(A1) the $p$-value $p_p$ of the “positive meta class” can be considered as an approximation of the $p$-value $p_y$ of the class $y$ when assigned to the instance $x$; and

(A2) the $p$-value $p_n$ of the “negative meta class” can be considered as an approximation of the sum of $p$-values of the classes $Y \setminus \{y\}$ when assigned to the instance $x$.

The meta-conformity approach is based on the meta-conformity assumptions (A1) and (A2). Given an instance $x \in X$ to be classified, the classification $y \in Y$ provided by the base classifier $B$ for $x$, and the $p$-values $p_p$ and $p_n$ of the “positive meta class” and “negative meta class” provided by the meta classifier $M$ for $x$, the approach approximates for the combined classifier $B:M$ the $p$-value $p_y$ of the class $y$ using
the $p$-value $p_p$ of the “positive meta class” and the sum of the $p$-values of the classes $Y \setminus \{y\}$ using the $p$-value $p_n$ of the “negative meta class”.

The meta-conformity approach allows considering the combined classifier $B:M$ as a scoring classifier with score ratio $\frac{p_n}{p_p}$ on the instances to be classified [13]. Thus, to decide whether a classification provided by the base classifier $B$ for an instance $x$ is reliable we need to determine a reliability threshold $T$ on the score ratio $\frac{p_n}{p_p}$ (see Task 2 from Section 1). In this way, if the score ratio $\frac{p_n}{p_p}$ that corresponds to $x$ is greater than $T$, then the classification of $x$ is considered as reliable; otherwise, is considered as unreliable and the instance $x$ is left unclassified. Hence, each combined classifier $B:M$ has to be specified by a reliability threshold $T$ on the score ratio $\frac{p_n}{p_p}$ that corresponds to the concrete task for reliable classification. We note that the threshold $T$ imposes a certain accuracy (on the classified instances) and rejection of the combined classifier $B:M$ (see Section 3).

6. Constructing combined classifiers with predefined generalization performance

Given a base classifier $B$ and a meta conformity-based classifier $M$, we can construct combined classifier $B:M$ with predefined target accuracy rate $At_{B:M}$ on the instances that $B:M$ can classify. The key idea is as follows. To decide whether instance classifications of the combined classifier $B:M$ are reliable, we have to obtain a reliability threshold $T$ on the scores $\frac{p_n}{p_p}$. The threshold $T$ has to be chosen so that the accuracy rate of the combined classifier $B:M$ equals the target accuracy rate $At_{B:M}$. To obtain the threshold $T$ we note that in the meta-conformity approach the $p$-values of the combined classifier $B:M$ equal the $p$-values of the meta conformity-based classifier $M$. By Theorem 2 the accuracy rate $A_{B:M}$ of the combined classifier $B:M$ equals the precision rate $P_M$ of the meta conformity-based classifier $M$. Thus, we can obtain the threshold $T$ as a threshold on the $p$-values of the meta conformity-based classifier $M$ so that the precision rate $P_M$ of $M$ becomes equal to the target accuracy rate $At_{B:M}$.

To build a meta conformity-based classifier $M$ with target precision rate $Pt_M$ we propose a ROC-based procedure for classifier construction. The procedure is based on our results presented in [23]. It constructs the classifier $M$ with the target precision rate $Pt_M$ by identifying the necessary reliability threshold $T$. This is done in three steps:

1. construct the ROC convex hull (ROCCH) of the meta conformity-based classifier $M$ using the score ratio $\frac{p_n}{p_p}$, where $p_p$ is the $p$-value of the “positive meta class” and $p_n$ is the $p$-value of the “negative meta class” (see Fig. 6). Constructing ROCCH is realized using an internal $k$-fold validation process similar to that described in Subsection 5.1.

2. construct the iso-precision line of the target precision rate $Pt_M$ given by the equation $TPr_M = \frac{Pt_M N_M}{Pr_M N_M} = \frac{Pr_M N_M}{Pr_M N_M} PPr_M$, where $P_M$ is the number of positive meta instances and $N_M$ is the number of negative meta instances. We note that this line consists of points representing classifiers with the target precision rate $Pt_M$. This property explains the next step of the procedure.

3. determine the reliability threshold $T$ as the value of the ratio $\frac{p_n}{p_p}$ in the intersection point $I$ of the ROCCH and the iso-precision line (see Fig. 6).

The precision rate $P_M$ of the meta conformity-based classifier $M$ with the reliability threshold $T$ will be equal to the target precision rate $Pt_M$ [23]. Thus, by Theorem 2 the accuracy rate $A_{B:M}$ of the combined classifier $B:M$ on the classified instances will be equal to the precision rate $Pt_M$ of the meta classifier $M$; i.e., it will be equal to the target accuracy rate $At_{B:M}$.

By Theorem 2 the accuracy rate $A_{B:M}$ of the combined classifier $B:M$ is maximized when the precision rate $P_M$ of the meta classifier $M$ is maximized. This precision rate is maximized for the iso-precision
Fig. 6. ROCCH of the conformity-based nearest-neighbor classifier \([17]\) trained on meta data obtained by a naive Bayes classifier on the Wisconsin breast-cancer data \([1]\). ROCCH is intersected by iso line of 0.9 precision rate.

line with slope equal to the slope of the line segment of the ROCCH starting from the origin. The highest point of this segment maximizes the number of covered meta instances (see point \((0, 0.74)\) of ROCCH in Fig. 6). Thus, in this point by Theorem 2 the accuracy rate of the combined classifier \(B:M\) is maximized while the rejection rate is minimized.

7. Experiments

To test the meta-conformity approach we have first designed a prototype of the combined classifiers. Then, we have experimented with combined classifiers that match the prototype on 12 UCI data multisets \([1]\).

The prototype of the combined classifiers has been designed to meet the requirements (S1) and (S2) (see Subsection 5.3). To satisfy the requirement (S1) we have chosen the naive-Bayes classifier \((NB)\) \([15]\) for the base classifier, since it does not compute classification \(p\)-values.\(^4\) To satisfy the requirement (S2) we have chosen the conformity-based nearest-neighbor classifier \((CNN)\) for the meta classifier (see Section 4), since it is a non-linear classifier and it does compute classification \(p\)-values. The resulting combined

\(^4\)The “naive” feature independence assumption is often violated in practice. Thus, the posterior class probabilities provided by \(NB\) are often not calibrated \([25]\).
classifiers have been denoted by \(NB:CNN\). The metadata for the meta classifiers \(CNN\) in \(NB:CNN\) have been generated in an internal 10-fold cross-validation process (see Fig. 4). The classification \(p\)-values of \(NB:CNN\) have been formed according to the meta-conformity approach.

We have experimented with the combined classifiers \(NB:CNN\) on 12 UCI data multisets \([1]\) by varying the reliability threshold \(T\) in the interval \([0, +\infty)\). If the ratio \(p_n/p_m\) for an instance has been greater than \(T\), the classification has been considered as reliable; otherwise, the classification has been considered as unreliable and the instance has been left unclassified. For each value of the reliability threshold \(T\) we have evaluated the accuracy rate and rejection rate using the 10-fold cross-validation method. We present the results as accuracy/rejection graphs of the combined classifiers \(NB:CNN\) in Fig. 7. Each point in the graphs represents a \(NB:CNN\) classifier with certain reliability threshold \(T\). The left most points in the graphs represent \(NB:CNN\) classifiers with thresholds \(T\) equal to 0. The next points represent \(NB:CNN\) classifiers with increasing thresholds \(T\) in the range \((0, +\infty)\). The right most points in the graphs represent \(NB:CNN\) classifiers with thresholds \(T\) approaching \(+\infty\). The rejection rates of the combined classifiers \(NB:CNN\) for which the accuracy rate of 1.0 has been achieved are presented in Table 1.

To demonstrate our ROC procedure for constructing combined classifiers with predefined generalization performance we have applied the procedure for the same UCI data multisets \([1]\) by varying the reliability threshold \(T\) in the interval \([0, +\infty)\). If the ratio \(p_n/p_m\) for an instance has been greater than \(T\), the classification has been considered as reliable; otherwise, the classification has been considered as unreliable and the instance has been left unclassified. For each value of the reliability threshold \(T\) we have evaluated the accuracy rate and rejection rate using the 10-fold cross-validation method. We present the results as accuracy/rejection graphs of the combined classifiers \(NB:CNN\) in Fig. 7. Each point in the graphs represents a \(NB:CNN\) classifier with certain reliability threshold \(T\). The left most points in the graphs represent \(NB:CNN\) classifiers with thresholds \(T\) equal to 0. The next points represent \(NB:CNN\) classifiers with increasing thresholds \(T\) in the range \((0, +\infty)\). The right most points in the graphs represent \(NB:CNN\) classifiers with thresholds \(T\) approaching \(+\infty\). The rejection rates of the combined classifiers \(NB:CNN\) for which the accuracy rate of 1.0 has been achieved are presented in Table 1.

To demonstrate our ROC procedure for constructing combined classifiers with predefined generalization performance we have applied the procedure for the same UCI data multisets. Constructing ROCCH has been realized using an internal 10-fold cross-validation process similar to that of the meta data generation (see Subsection 5.1). The reliability thresholds \(T\) have been formed to generate combined classifiers \(NB:CNN\) for the case when target accuracy rate \(At\) of 1.0 is required. The accuracy and rejection rates of the combined classifiers \(NB:CNN\) for those thresholds \(T\) are presented in Table 2. The table shows that the accuracy rates are close to the target accuracy rate of 1.0. The maximum deviation is 0.03. We explain this effect with the ROCCH of the conformity-based meta classifiers \(M\) that can be imprecisely constructed in some cases. This can be due to the size of the training (meta) data and/or the well-known instability of the 10-fold cross-validation process for the meta data generation. In fact the largest deviations we observe for relatively small data with size less than 300 instances: Audiology, Hepatitis, Heart-Statlog, Iris, Lymphography, and Zoo. Thus, we may conclude that our ROC procedure for classifier construction is applicable in practice, but it produces more exact results when the size of the data grows.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>(R_{NB:CNN})</th>
<th>(R_{CNN})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annealing</td>
<td>0.57*</td>
<td>0.88</td>
</tr>
<tr>
<td>Audiology</td>
<td>0.84*</td>
<td>0.99</td>
</tr>
<tr>
<td>Wisc. breast cancer</td>
<td>0.29</td>
<td>0.31</td>
</tr>
<tr>
<td>Glass</td>
<td>0.82*</td>
<td>0.88</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>0.71</td>
<td>0.65</td>
</tr>
<tr>
<td>Heart-Statlog</td>
<td>0.94</td>
<td>0.85*</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>0.70</td>
<td>0.55*</td>
</tr>
<tr>
<td>Iris</td>
<td>0.29</td>
<td>0.13*</td>
</tr>
<tr>
<td>Lymphography</td>
<td>0.80*</td>
<td>0.99</td>
</tr>
<tr>
<td>Soybean</td>
<td>0.87*</td>
<td>0.90</td>
</tr>
<tr>
<td>Vote</td>
<td>0.52</td>
<td>0.47*</td>
</tr>
<tr>
<td>Zoo</td>
<td>0.23</td>
<td>0.14*</td>
</tr>
</tbody>
</table>
Fig. 7. The accuracy/rejection graphs of NB-CNN (- - -) and CNN (—).
Table 2

<table>
<thead>
<tr>
<th>Data Set</th>
<th>$R_{NB,CNN}$</th>
<th>$A_{NB,CNN}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annealing</td>
<td>0.65</td>
<td>1.00</td>
</tr>
<tr>
<td>Audiology</td>
<td>0.83</td>
<td>0.97</td>
</tr>
<tr>
<td>Wisc. breast cancer</td>
<td>0.29</td>
<td>1.00</td>
</tr>
<tr>
<td>Glass</td>
<td>0.87</td>
<td>1.00</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>0.71</td>
<td>0.98</td>
</tr>
<tr>
<td>Heart-Statlog</td>
<td>0.88</td>
<td>0.97</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>0.72</td>
<td>1.00</td>
</tr>
<tr>
<td>Iris</td>
<td>0.31</td>
<td>0.99</td>
</tr>
<tr>
<td>Lymphography</td>
<td>0.79</td>
<td>0.98</td>
</tr>
<tr>
<td>Soybean</td>
<td>0.72</td>
<td>1.00</td>
</tr>
<tr>
<td>Vote</td>
<td>0.49</td>
<td>1.00</td>
</tr>
<tr>
<td>Zoo</td>
<td>0.18</td>
<td>0.99</td>
</tr>
</tbody>
</table>

8. Comparison

We compare the combined classifiers $NB:CNN$ with the pure classifiers $CNN$ trained on the same 12 UCI data multisets. The accuracy/rejection graphs of $CNN$ classifiers have similarly been generated and are presented in Fig. 7. A comparison is provided in Table 1 showing the rejection rates of the classifiers $NB:CNN$ and $CNN$ for which accuracy rate of 1.0 has been achieved. The comparison shows that each of these classifiers is significantly better for five data multisets. This means that both classifiers are good and are different. From this result we may conclude that the meta-conformity approach approximates well the classification $p$-values. Thus, the meta-conformity assumptions (A1) and (A2) on which the approach is based do hold in practice.

9. Conclusion

In this paper we have proposed a meta-conformity approach to allow the conformity framework to be applied to any classifier. The approach does not put any restriction on the type of the classifier, but it requires the availability of a non-linear conformity-based classifier for the meta level. The experiments have shown that the meta-conformity approach approximates well classification $p$-values. In addition, the approach allows constructing classifiers with predefined generalization performance.

Future research will concentrate on two main research directions: further approach development and new applications. When we consider the further approach development we note one shortcoming of the meta-conformity approach, namely that it approximates only the $p$-value of the winning class and the sum of the $p$-values of the remaining classes provided by the base classifier; i.e., it is not capable of providing $p$-value for each possible class. This is due to the fact that the meta classifier predicts correctness of the winning class of the base classifier; i.e., the class with the highest score computed by that classifier. To provide $p$-value for each possible class we note that the meta-conformity approach can be easily modified: instead of learning one meta classifier we can learn a cascade of meta-classifiers so that the $i$-th meta-classifier predicts whether the class with the $i$-th highest score provided by the base classifier is the correct one. Thus, assuming that the meta classifiers are conformity-based, our modified meta-conformity approach will be able to approximate $p$-values for all possible classes.
This paper has so far shown that the meta-conformity approach can be successfully applied to artificial classifiers. The open research question is whether we can apply our approach to human experts that perform classification tasks. If the answer will be positive, the meta-conformity approach will be able to provide p-values for the classifications of the human experts, a research direction that has not been studied.

Acknowledgement

We would like to thank anonymous reviewers for their constructive advices and comments.

References


