Polychotomous Classification With Pairwise Classifiers: a New Voting Principle

Florin Cutzu

Dept. of Computer Science
Indiana University
Bloomington, IN 47405
USA
florin@indiana.edu

Abstract. A new principle for performing polychotomous classification with pairwise classifiers is introduced: if pairwise classifier $\mathcal{N}_{ij}$, trained to discriminate between classes i and j, responds "i" for an input $\boldsymbol{x}$ from an unknown class (not necessarily i or j), one can at best conclude that $\boldsymbol{x} \notin j$. Thus, the output of pairwise classifier $\mathcal{N}_{ij}$ can be interpreted as a vote against the losing class j, and not, as existing methods propose, as a vote for the winning class i. Both a discrete and a continuous classification model derived from this principle are introduced.

1 Introduction

1.1 Problem Statement

Consider the problem of m-way classification ($m \geq 2$) of the random variable $\boldsymbol{x} \in \mathcal{R}^d$. The probability density function of $\boldsymbol{x}$ is a mixture of m partially overlapping components corresponding to the classes of interest in the classification problem: $p(\boldsymbol{x}) = \sum_1^m w_k p(\boldsymbol{x} \mid k)$ where the weights are the prior class probabilities: $w_k = P(k)$, $\sum_1^m w_k = 1$. The training data consists of n input vectors $\boldsymbol{x}_i$, which usually are typical representatives of the classes, and the corresponding class labels $y_i \in \{1, \ldots, m\}$. The goal is to assign class labels to novel inputs.

The direct approach is to use a single m-way classifier; an alternative approach is to employ several k-way classifiers ($k < m$) and combine their outputs in a m-way classification decision. In this paper the case $k = 2$ (pairwise classifiers) is explored.

1.2 Current Approaches

The idea of performing complex multi-way classification tasks by combining multiple simpler, specialized classifiers is not new and various approaches based on this principle have been presented in the research literature as well as in pattern recognition textbooks such as [6].

Of particular interest in the context of this paper are Friedman's method [2], the pairwise coupling model of Hastie & Tibshirani [3] and the related approach presented in [6].

Friedman's Voting Method. The Bayesian solution to the classification of $\boldsymbol{x}$ requires that estimates of the m class posterior probability densities $P(k \mid \boldsymbol{x})$ be obtained in the training stage.

Friedman's approach reformulates the Bayesian solution to reduce the m-way classification problem to $m(m-1)/2$ pairwise discrimination problems, as follows. During training, possibly using neural network techniques, estimates of the following ratios of probability densities are obtained:

$$r_{ij}(\boldsymbol{x}) = \frac{P(i \mid \boldsymbol{x})}{P(j \mid \boldsymbol{x}) + P(i \mid \boldsymbol{x})} \quad (i, j = 1, \ldots, m) \tag{1.1}$$

The functions r_{ij} learned in the training phase are then used in the testing phase. If for input vector $\boldsymbol{x}$, $r_{ij}(\boldsymbol{x}) > 0.5$ then $P(i \mid \boldsymbol{x}) > P(j \mid \boldsymbol{x})$, and class i is the "winner" of the $i - j$ comparison. Thus, the output of pairwise estimator (i, j) can be interpreted as a vote for either class i or for class j. There are $m(m-1)/2$ non-trivially distinct functions r_{ij}, one for each pair of classes. Input vector $\boldsymbol{x}$ is assigned to class k if class k wins the most votes, or two-class classifier decisions $k - i$ $(i = 1, \ldots, m)$.

Pairwise Coupling and Similar Models. The pairwise coupling model [3] assumes that, for input $\boldsymbol{x}$, the output of pairwise classifier (i, j) is given by:

$$s_{ij}(\boldsymbol{x}) = P(i \mid \boldsymbol{x}, i \vee j) \tag{1.2}$$

and that

$$s_{ij}(\boldsymbol{x}) = \frac{P(i \mid \boldsymbol{x})}{P(i \mid \boldsymbol{x}) + P(j \mid \boldsymbol{x})} \tag{1.3}$$

which is the same relation as in Eq. 1.1. The authors then proceed to find the set of probabilities $P(i \mid \boldsymbol{x})$ that best fit the set of classier outputs s_{ij} via Eq. 1.3. Thus, while Friedman determines only the class with maximum posterior probability, the pairwise coupling model estimates the class posterior probabilities of all classes.

A similar model is given in [6]. For input $\boldsymbol{x}$, the output of pairwise classifier (i, j) is given, as in the pairwise coupling model, by:

$$s_{ij}(\boldsymbol{x}) = P(i \mid \boldsymbol{x}, i \vee j)$$

By Bayes law

$$s_{ij}(\boldsymbol{x}) = \frac{w_i P(\boldsymbol{x} \mid i)}{w_i P(\boldsymbol{x} \mid i) + w_j P(\boldsymbol{x} \mid j)} \tag{1.4}$$

where w_i is the prior probability of class i. Note that, as opposed to Eq. 1.3 of the pairwise coupling model, this is an exact relation.

Given the classifier outputs s_{ij}, using Eq. 1.4 and Bayes' law

$$P(i \mid \boldsymbol{x}) = \frac{w_i P(\boldsymbol{x} \mid i)}{\sum_{j=1}^{m} w_j P(\boldsymbol{x} \mid j)}$$

one can compute all class posterior probabilities.

More Complex Schemes. The literature contains a relatively large number of papers on the problem of combining simpler classifiers into multi-way classifiers: [1, 4, 5, 7]. These papers will not be discussed any further, since they generalize the problem and its solution in various ways, but, as far as the focus of the present paper is concerned, do not fundamentally change the principles introduced by Friedman.

2 The Proposed Approach

2.1 Problems with Current Approaches

Friedman's method, being a form of Bayesian classification, requires estimates of the probabilities of the various classes at the input vector. Unfortunately, class probability estimation is a difficult problem. Thus, it is desirable to design a classification method that retains the advantage of reducing the multi-way problem to a set of two-way decisions but does not require pairwise class density comparison.

The problem with with the pairwise coupling and related models is more basic. According to these models, the output of classifier (i, j) for input vector $\boldsymbol{x}$ is given by $s_{ij}(\boldsymbol{x}) = P(i \mid \boldsymbol{x}, i \vee j)$. In other words, the input vector is assumed to belong to class i or to class j. However, in the testing phase, it is impossible to ensure that input vector $\boldsymbol{x}$ fed to classifier (i, j) belongs to either class i or j. If the input belongs to some other class, $\boldsymbol{x} \in k \neq i, j$, then the output of classifier (i, j) can no longer be interpreted as in Eqs 1.2, 1.3, 1.4, and these models cannot be applied.

2.2 "Non-probabilistic" Pairwise Classifiers

The goal of this paper is to formulate a multi-way classification scheme based on pairwise classifiers that do not estimate the pairwise class probability ratios of the type 1.1 for the input vector. Such classifiers will be hereafter termed "non-probabilistic". A typical example of such a classifier is the multilayer perceptron.

A non-probabilistic classifier performs certain calculations on the components of the input vector. For example, such a pairwise classifier may discriminate between two classes by comparing a certain continuous feature with a threshold, or by detecting the presence of one of two distinguishing features, or by performing more complex calculations on the vector components in the case of the neural networks. In vision application, one may differentiate between two image classes by detecting one or two image templates at certain positions in the image.

In the training stage, using neural networks or related techniques as pairwise classifiers is attractive because, in general, learning one m-way classification is more expensive computationally than learning $m(m-1)/2$ two-way classifications.

For input $\boldsymbol{x} \in i \vee j$, the output of a pairwise neural network $\mathcal{N}_{i,j}$ trained to discriminate between classes i and j (by outputting 0 for class i and 1 for class

j) can be *interpreted* as deciding whether $p_i(\boldsymbol{x}) > p_j(\boldsymbol{x})$ (despite the fact that such networks do not actually estimate class probabilities).

One is therefore tempted to try to apply Friedman's voting method to the outputs of neural, non-probabilistic pairwise classifiers. However, the output of pairwise network $\mathcal{N}_{ij}(\boldsymbol{x})$ has the same meaning as $r_{ij}(\boldsymbol{x})$ in Eq. 1.1 only if the input $\boldsymbol{x} \in i \vee j$, condition that can be verified only for the training set.

This is a problem, since in practical classification problems the different classes have finite extents and overlap only partially in input space. Consequently, there usually exist regions in input space where only one single class is present, and consequently the ratio $r_{ij}(\boldsymbol{x})$ may be undefined, since both $P_i(\boldsymbol{x})$ and $P_j(\boldsymbol{x})$ may be 0. At these locations the neural network pairwise classifier $\mathcal{N}_{i,j}$ will have some output in $[0, 1]$; however, comparing this output to 0.5 to determine whether $p_i > p_j$ is no longer legitimate. The meaning of the output of $\mathcal{N}_{ij}(\boldsymbol{x})$ for $\boldsymbol{x} \notin i \vee j$ is not obvious, and applying Friedman's voting scheme is not justified.

2.3 A Novel Voting Principle: Vote Against, Not For

To correctly use the pairwise non-probabilistic classifiers $\mathcal{N}_{ij}$ for classification of novel inputs one must interpret the outputs of these classifiers for inputs from untrained-for classes $k \neq i, j$.

The problem with directly applying the Friedman [2] or Hastie-Tibshirani [3] approaches to non-probabilistic pairwise classifiers is the following. Consider classifier $\mathcal{N}_{ij}$, trained to discriminate between classes i and j. If, for input $\boldsymbol{x}$ of unknown class membership, the output of the classifier is "i", applying the Friedman algorithm results in a vote for class i. Similarly, the Hastie-Tibshirani algorithm increases the probability of class i.

However, since the true class of $\boldsymbol{x}$ can be other than i or j, such an interpretation of the output of classifier $\mathcal{N}_{ij}$ can result in *false positive* errors–i.e., falsely attributing $\boldsymbol{x} \notin i \cup j$ to class i or j. On the other hand, one can expect that, if properly trained, the pairwise classifiers do not make (many) *false negative* errors—i.e., give the wrong response to inputs from trained-for classes. Thus, false positive errors are much more likely than false negative errors.

This observation leads to the following classification rule. If, for input $\boldsymbol{x}$ of unknown class membership, classifier $\mathcal{N}_{ij}$ responds "i", one can *not* conclude that $\boldsymbol{x} \in i$. Due to the possibility of false positive errors, one can only conclude that $\boldsymbol{x} \notin j$, because, if the input was from class j, the classifier $\mathcal{N}_{ij}$ would have responded "j" (assuming no false negative errors). In other words, one votes *against* j, not *for* i.

Formally, $(\boldsymbol{x} \in i \rightarrow \mathcal{N}_{ij}$ *responds "i"*$) \equiv (\boldsymbol{x} \notin i \rightarrow \mathcal{N}_{ij}$ *does not respond "i"*$) \not\equiv (\mathcal{N}_{ij}$ *responds "i"* $\rightarrow \boldsymbol{x} \in i)$.

2.4 A Model for the Class Posterior Probabilities

Let y_{ij} denote the output of non-probabilistic pairwise classifier $\mathcal{N}_{ij}$ for an input $\boldsymbol{x}$ from an arbitrary class. y_{ij} is a random variable. Using solely the principle

formulated above, a model for the class posterior probabilities $P(k \mid y_{ij})$ can be formulated, as follows. This model represents a conservative (maximum ignorance) interpretation of the outputs of the classifiers when nothing is known about the class membership of the input.

Discrete Output Classifiers. To simplify, assume first the classifiers $\mathcal{N}_{ij}$, $i, j = 1, \ldots, m$ output binary decisions: either i or j.

Given that classifier $\mathcal{N}_{ij}$ outputs j, what are the probabilities of each of the m classes?

It can reasonably be assumed that the classifiers are properly trained, and thus very few false negative errors occur. Therefore, if classifier N_{ij} outputs j, the posterior probability of class i is reduced to zero, or more generally, to a very small fraction ϵ_{ji} of its prior probability w_i, that is, $P(i \mid y_{ij} = j) = \epsilon_{ji} w_i$.

All the other classes $k \neq i$ (not only class j!) are possible, it is hypothesized, with probabilities that sum up to $1 - \epsilon_{ji} w_i$ and are in ratios equal to the ratios of their prior probabilities w_k.

Therefore:

$$P(i \mid y_{ij} = j) = \epsilon_{ji} w_i; \tag{2.1}$$

$$P(j \mid y_{ij} = j) = \frac{w_j (1 - \epsilon_{ji} w_i)}{1 - w_i}; \tag{2.2}$$

$$\text{Generally, } \forall k \neq i : P(k \mid y_{ij} = j) = \frac{w_k (1 - \epsilon_{ji} w_i)}{1 - w_i}; \tag{2.3}$$

As desired, $\forall k, h \neq i$, $P(k \mid y_{ij} = j)/P(h \mid y_{ij} = j) = w_k/w_h$. As required, $\sum_{k=1}^{m} P(k \mid \mathcal{N}_{ij} = j) = 1$.

The factor ϵ_{ji} determines the probability that classifier $\mathcal{N}_{ij}$ outputs "i" for an input of class j, and it can be estimated in the training stage. Similarly, ϵ_{ij} measures the probability that classifier $\mathcal{N}_{ij}$ outputs "j" for an input of class i. If ϵ_{ji} is sufficiently small, $\forall k \neq i, P(k \mid y_{ij} = j) > w_k$.

Note that if w_i increases for w_j constant $P(j \mid y_{ij} = j)$ increases nonlinearly. This is a consequence of the fact that class i is excluded and the "missing probability" is made up for in part by class j. If $w_i + w_j =$ constant then $P(j \mid y_{ij} = j)$ decreases with increasing w_i.

It will be shown later that if the prior probabilities w_i are equal, and all ϵ_{ij} are equal, the voting-against method is equivalent to Friedman's voting-for method. Intuitively, voting against class j is equivalent to voting equally for each of the other classes $j \neq i$ — in other words, if $\mathcal{N}_{ij}$ classifier responds "i", the true class can be any class except j, with the same probability.

Continuous Output Classifiers. If classifier outputs y_{ij} are not binary but continuous in $[0, 1]$ a simple possibility is to interpolate between the limiting cases $P(k \mid y_{ij} = 0)$ and $P(k \mid y_{ij} = 1)$ to obtain $P(k \mid y_{ij} \in (0, 1))$. Assuming that for classifier $\mathcal{N}_{ij}$, $y_{ij} = 0$ corresponds to $y_{ij} = j$ and $y_{ij} = 1$ corresponds to

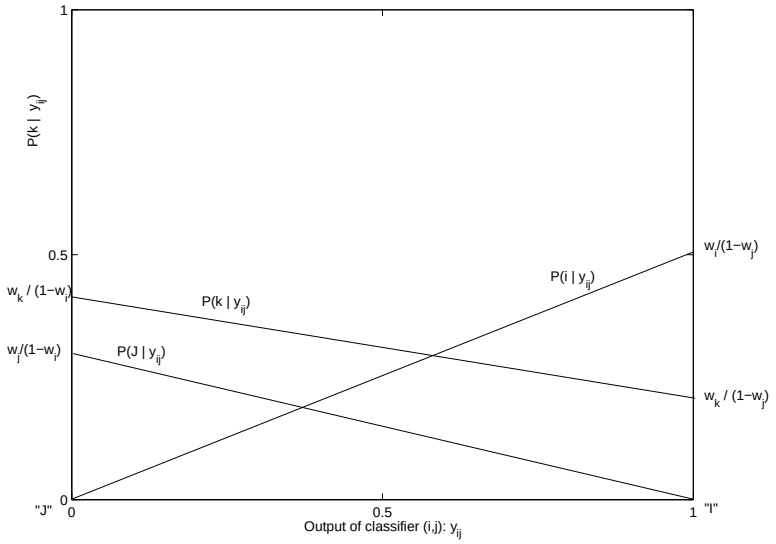

Fig. 1. A model for the probabilities of the m classes as functions of y_{ij}, the output of classifier $\mathcal{N}_{ij}$. For simplicity, the figure displays the case where the false negative rate is zero ($\epsilon = 0$).

$y_{ij} = i$, these limiting probabilities are given, for the various values of k, by Eqs 2.1, 2.2, 2.3. By linear interpolation:

$$P(k \mid y_{ij}) = (1 - y_{ij})\frac{w_k(1 - \epsilon_{ji}w_i)}{1 - w_i} + y_{ij}\frac{w_k(1 - \epsilon_{ij}w_j)}{1 - w_j},\ k \neq i, j \qquad (2.4)$$

$$P(i \mid y_{ij}) = (1 - y_{ij})\epsilon_{ji}w_i + y_{ij}\frac{w_i(1 - \epsilon_{ij}w_j)}{1 - w_j},$$

$$P(j \mid y_{ij}) = (1 - y_{ij})\frac{w_j(1 - \epsilon_{ji}w_i)}{1 - w_i} + y_{ij}\epsilon_{ij}w_j.$$

It can be verified that $\sum_{k=1}^{m} P(k \mid y_{ij}) = 1$. Figure 1 illustrates class posterior probabilities $P(k \mid y_{ij})$ conditional on the output of classifier $\mathcal{N}_{ij}$.

Determining the joint classifier class-posterior probability. The next step is to determine the joint $P(i \mid \boldsymbol{y})$, where $\boldsymbol{y} = [y_{1,2}, \ldots, y_{m-1,m}]$. By Bayes

$$p(y_{ij} \mid i) = \frac{P(i \mid p_{ij})p(y_{ij})}{w_i}$$

.

If the assumption is made that the outputs of the classifiers in the classifier bank are (conditionally) independent, the conditional joint probability $p(\boldsymbol{y} \mid i)$

of the output of the full classifier bank is:

$$p(\boldsymbol{y} \mid i) = \prod_{k,j} p(y_{kj} \mid i) = w_i^{-m(m-1)/2} \prod_{k,j} P(i \mid y_{kj}) p(y_{kj}).$$

By Bayes, the desired class posterior probability is:

$$P(i \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid i) w_i}{p(\boldsymbol{y})} = w_i^{1-m(m-1)/2} \frac{\prod_{k,j} P(i \mid y_{kj}) p(y_{kj})}{p(\boldsymbol{y})} \tag{2.5}$$

Given that $p(\boldsymbol{y})$ and $\prod_{k,j} p(y_{kj})$ are the same for all class posterior probabilities $P(i \mid \boldsymbol{y})$, $i = 1, \ldots, m$, they can be ignored. Therefore

$$P(i \mid \boldsymbol{y}) \sim w_i^{1-m(m-1)/2} \prod_{k,j} P(i \mid y_{kj}) \tag{2.6}$$

Taking logarithm:

$$\log P(i \mid \boldsymbol{y}) = c + (1 - m(m-1)/2) \log w_i + \sum_{k,j} \log P(i \mid y_{kj}) \tag{2.7}$$

where c is a constant that is the same for all classes i. The logarithm exists for $P(i \mid y_{kj}) > 0$, and this requires that $\epsilon_{ij} > 0$, condition which can always be met.

Discrete Case. For classifiers with *binary outputs* $y_{ij} = i$ or j, replacing the probabilities $P(i \mid y_{kj})$ with the expressions given in Eq. 2.1, 2.2, 2.3, gives:

$$\begin{gathered} \log P(i \mid \boldsymbol{y}) = c + \log w_i + \\ \textstyle\sum_{j \neq i} \log \left(\epsilon_{ji}, \text{ if } y_{ij} = j \ ; \ \frac{1-\epsilon_{ij} w_j}{1-w_j}, \text{ if } y_{ij} = i \right) + \\ \textstyle\sum_{k,j \neq i} \log \left(\frac{1-\epsilon_{kj} w_j}{1-w_j}, \text{ if } y_{kj} = k \ ; \ \frac{1-\epsilon_{jk} w_k}{1-w_k}, \text{ if } y_{kj} = j \right) \end{gathered} \tag{2.8}$$

This equation indicates that each pairwise classifier contributes "votes" for class i. If a classifier of type $\mathcal{N}_{ij}$ outputs j, its vote is negative ($\epsilon < 1$); otherwise the votes are positive ($(1-\epsilon w_j)/(1-w_j) > 1$), the strength of the vote depending on the network output, prior class probabilities, and the false negative error rates ϵ_{ij}.

If all classes have the same prior probability $w_i = 1/m$ and if all ϵ_{ij} are equal, then the classifiers of type $\mathcal{N}_{kj}$, $k, j \neq i$ become irrelevant to classifying class i, and the relation above is equivalent to Friedman's voting formula, which thus obtains as a special case:

$$\log P(i \mid \boldsymbol{y}) = c + \sum_{j \neq i} \log \left(\epsilon, \text{ if } y_{ij} = j \ ; \ \frac{m-\epsilon}{m-1}, \text{ if } y_{ij} = i \right) \tag{2.9}$$

Since $\log \frac{m-\epsilon}{m-1} > \log \epsilon$, the probability of class i increases with the proportion of the $m-1$ classifiers of type $\mathcal{N}_{ij}$ that respond i (as opposed to j).

Continuous Case. For continuous-output classifiers, using Eq. 2.4, and assuming that classifier $\mathcal{N}_{ij}$ outputs 1 for class i and 0 for class j:

$$\begin{gathered}\log P(i \mid \boldsymbol{y}) = c + \log w_i + \\ \sum_{j \neq i} \log \left(\epsilon_{ji}(1 - y_{ij}) + \tfrac{1-\epsilon_{ij} w_j}{1-w_j} y_{ij} \right) + \\ \sum_{k,j \neq i} \log \left(\tfrac{1-\epsilon_{kj} w_j}{1-w_j} y_{kj} + \tfrac{1-\epsilon_{jk} w_k}{1-w_k} (1 - y_{kj}) \right).\end{gathered} \tag{2.10}$$

This equation is the one of the main results of this paper.

If all classes have the same prior probability $w_i = 1/m$ and if all ϵ_{ij} are equal, then the classifiers of type $\mathcal{N}_{kj},\ k, j \neq i$ become irrelevant to classifying class i:

$$\log P(i \mid \boldsymbol{y}) = c + \sum_{j \neq i} \log \left(\epsilon(1 - y_{ij}) + \frac{m - \epsilon}{m - 1} y_{ij} \right) \tag{2.11}$$

which is a continuous analogue of the Friedman voting rule (soft voting).

2.5 Classifying with an incomplete pairwise classifier set

From a computational standpoint it is important to study the situation in which only a subset of all $m(m-1)/2$ pairwise classifiers are used.

Here only the simplest situation is considered, namely, binary output pairwise classifiers that do not make false negative errors ($\epsilon = 0$).

Consider a subset of the complete classifier set consisting of $b \leq m(m-1)/2$ classifiers, and consider class i. Assume there are $n(i) \leq (m-1)$ classifiers of type $\mathcal{N}_{i(*)}$, trained to discriminate class i from other classes $(*)$. Assume that for input $\boldsymbol{x}$ of indeterminate class membership, a number $v(i) \leq n(i)$ of these classifiers will respond "i". Because voting against class $j \neq i$ is equivalent to voting for all classes $k \neq j$, including i, the number of votes *for* class i resulting from the votes *against* classes $j \neq i$ is, for input $\boldsymbol{x}$:

$$f(i) = b - n(i) + v(i) \tag{2.12}$$

Therefore, the vote-against method results in $f(i)$ votes for class i; under usual voting-for method class i receives $v(i)$ votes. The voting-against and the voting-for methods are equivalent if $n(i)$ does not depend on i, and in particular if all $m(m-1)/2$ pairwise classifiers are used in the classification process.

The voting-against method is superior if not all classifiers are used. Unlike the voting-for method, regardless of the number of pairwise classifiers employed, the voting-against method never selects the wrong class: it never casts a vote against the true class (assuming no false negative errors occur). If $\boldsymbol{x} \in i$ is input to classifier $\mathcal{N}_{ij}$, the vote-for method will correctly vote for class i; the vote-against method will, also correctly, vote against class j. However, if $\boldsymbol{x} \in i$ is input to classifier $\mathcal{N}_{kj}$, the vote-for method will incorrectly vote for either class k or j; the vote-against method will correctly vote against either class j or k. The vote-for method fails if the classifiers trained on the true class are not used

in the classification process; the vote-against method correctly selects the true class even if none of these are used. Both methods give the same, correct results if only the classifiers trained on the true class are used. The vote-against method, while never voting against the true class i, can however fail to cast votes against some classes $j \neq i$, resulting in a non-unique solution. However, this happens if the only classifiers used are a subset of the classifiers $\mathcal{N}_{ij}$ trained on the true class, i.

3 Summary and Conclusions

This paper addresses the problem of polychotomous classification with pairwise classifiers. The essential difference from previous methods such as [2] and [3] is that the pairwise classifiers considered in the present paper are common classifiers such as the multilayer perceptron which do not require class probability estimation to perform classification.

To handle such classifiers the paper introduces a new, conservative interpretation of the output of a pairwise classifier for inputs of unknown class membership. The observation at the basis of the proposed method is that, while an adequately-trained classifier will inevitably falsely recognize inputs from unknown classes, it will not fail to recognize inputs from trained-for classes. This approach has not only the theoretical advantage of being logically correct, but is also, unlike other methods, robust to reducing the number of pairwise classifiers used in the classification process.

Interpreting the output of a pairwise, or more generally, n-way classifier as evidence against (rather than for) a trained-for class has the advantage that it allows, conceptually, the classification of an input in an "unknown" class if there is evidence against all known classes.

Two practical classification models based on this principle were proposed: one for discrete-output pairwise classifiers, in Equation 2.8 and another for continuous-output classifiers, in Equation 2.10. The Friedman voting scheme is as a particular case of the proposed model.

In practice it should be possible to use a less conservative interpretation of the outputs of the pairwise classifiers. The idea is to use the training data not only for training the pairwise classifiers, but also for testing them and getting an idea, in probabilistic terms, of their behavior for inputs from untrained-for classes.

References

[1] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. In *Proc. 17th International Conf. on Machine Learning*, pages 9–16. Morgan Kaufmann, San Francisco, CA, 2000.

[2] Jerome H. Friedman. Another approach to polychotomous classification. Technical report, Stanford University, 1996.

[3] Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.

[4] Eddy Mayoraz and Ethem Alpaydin. Support vector machines for multi-class classification. In *IWANN (2)*, pages 833–842, 1999.

[5] Volker Roth. Probabilistic discriminative kernel classifiers for multi-class problems. *Lecture Notes in Computer Science*, 2191:246–266, 2001.

[6] Jürgen Schürmann. *Patern Classification. A Unified View of Statistical and Neural Principles.* John Wiley & Sons, Inc, New York, NA, 1996.

[7] B. Zadrozny. Reducing multiclass to binary by coupling probability estimates, 2001.