Pattern Recognition 43 (2010) 128 – 142

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Combining predictions in pairwise classification: An optimal adaptive voting strategy and its relation to weighted voting

Eyke Hüllermeier[a,*], Stijn Vanderlooy[b]

[a]Department of Mathematics and Computer Science, Marburg University, Germany
[b]Department of Knowledge Engineering, Maastricht University, The Netherlands

ARTICLE INFO

Article history:
Received 24 October 2008
Received in revised form 6 March 2009
Accepted 20 June 2009

Keywords:
Learning by pairwise comparison
Label ranking
Aggregation strategies
Classifier combination
Weighted voting
MAP prediction

ABSTRACT

Weighted voting is the commonly used strategy for combining predictions in pairwise classification. Even though it shows good classification performance in practice, it is often criticized for lacking a sound theoretical justification. In this paper, we study the problem of combining predictions within a formal framework of label ranking and, under some model assumptions, derive a generalized voting strategy in which predictions are properly adapted according to the strengths of the corresponding base classifiers. We call this strategy adaptive voting and show that it is optimal in the sense of yielding a MAP prediction of the class label of a test instance. Moreover, we offer a theoretical justification for weighted voting by showing that it yields a good approximation of the optimal adaptive voting prediction. This result is further corroborated by empirical evidence from experiments with real and synthetic data sets showing that, even though adaptive voting is sometimes able to achieve consistent improvements, weighted voting is in general quite competitive, all the more in cases where the aforementioned model assumptions underlying adaptive voting are not met. In this sense, weighted voting appears to be a more robust aggregation strategy.

1. Introduction

Learning by pairwise comparison (*aka* all-pairs, one-against-one, round robin learning, and pairwise classification) is a well-known decomposition technique which allows one to transform a multi-class classification problem, i.e., a problem involving $m > 2$ classes, into a number of *binary* classification problems [1]. To this end, a separate model is trained for each pair of class labels. Typically, the technique produces more accurate models than the alternative one-against-rest decomposition, which learns one model for each class using the examples of this class as positive examples and all others as negative examples. Also, despite the need to train a quadratic instead of a linear number of models, pairwise classification is computationally not more complex than one-against-rest. The reason is that the binary classification problems not only contain fewer training examples (because all examples that do not belong to either of the two classes are ignored), but the decision boundaries for each of the problems may also be considerably simpler than for the problems generated by the one-against-rest technique [2,1,3]. Both techniques are special cases of a more general approach that uses error correcting output codes to decompose a multi-class problem into several binary classification problems [4].

A critical step in pairwise learning is the aggregation of the predictions from the ensemble of binary models into a final classification. A large number of strategies have been proposed for this purpose, see for example [5–9]. Since the aggregation problem also occurs in all other decomposition methods and in ensemble methods, these research areas as well provide a large number of aggregation strategies (sometimes called classifier combination schemes); see for example [10] and references therein. However, since the semantics of these problems are different, we note that the aggregation strategies from different fields are not always interchangeable. In this paper, we solely focus on aggregating predictions from binary models learned by pairwise comparison. Judging from the relevant research literature, it is clear that the most commonly used aggregation strategy in practice is the simple *weighted voting*. In this strategy, the prediction of each binary model is counted as a weighted "vote" for a class label, and the class with the highest sum of votes is predicted. Even though weighted voting performs very well in practice, it is often criticized as being ad-hoc to some extent and for lacking a sound theoretical basis [11,12].

In this regard, the current paper makes the following three contributions. First, we propose a formal framework in which the

* Corresponding author.
E-mail addresses: eyke@mathematik.uni-marburg.de (E. Hüllermeier), s.vanderlooy@micc.unimaas.nl (S. Vanderlooy).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.06.013

aforementioned aggregation problem for pairwise learning can be studied in a convenient way. This framework is based on the setting of *label ranking* which has recently received attention in the machine learning literature [13–15,11]. Second, within this framework, we develop a new aggregation strategy called *adaptive voting*. This strategy allows one to take the strength of individual classifiers into consideration and, under certain assumptions, is provably optimal in the sense that it yields a MAP prediction of the class label (i.e., it predicts the label with maximum posterior probability).[1] Third, we show that weighted voting can be seen as an approximation of adaptive voting and, hence, approximates the MAP prediction. This theoretical justification is complemented by arguments suggesting that weighted voting is quite robust toward incorrect predictions of the binary models. Finally, all these results are confirmed by strong empirical evidence showing that adaptive voting is indeed able to outperform weighted voting in a consistent way, albeit by a very small margin. The experimental results also show that the superiority of adaptive voting only holds as long as its underlying model assumptions are (approximately) met by the ensemble of binary models. If these assumptions are strongly violated, weighted voting is at least competitive and, in this sense, appears to be more robust than adaptive voting.

The remainder of the paper is organized as follows. In Section 2, we review learning by pairwise comparison. In Section 3, we present the setting of label ranking and we argue for analyzing learning by pairwise comparison in this setting. We introduce the adaptive voting strategy and its underlying formal model in Section 4. In Section 5, we provide experiments with synthetic data. In Section 6, we focus on weighted voting and show that it can be seen as an approximation of adaptive voting. Experimental results on benchmark data sets are presented in Section 7. The paper ends with concluding remarks and future work in Section 8.

2. Learning by pairwise comparison

Learning by pairwise comparison is a popular decomposition technique that transforms an m-class classification problem, $m>2$, into a number of binary problems [1]. To this end, a separate model or base classifier $\mathscr{M}_{ij}$ is trained for each pair of labels $(\lambda_i,\lambda_j)\in\mathscr{L}\times\mathscr{L}$, $1\leq i<j\leq m$; thus a number of $m(m-1)/2$ models is needed in total (see Fig. 1). The base classifier $\mathscr{M}_{ij}$ is intended to discriminate instances with class label λ_i from those having class label λ_j.

The binary base classifier $\mathscr{M}_{ij}$ is trained using examples of classes λ_i and λ_j only. More specifically, an example $(\boldsymbol{x},\lambda_{\boldsymbol{x}})$, i.e., an instance $\boldsymbol{x}$ belonging to class label $\lambda_{\boldsymbol{x}}=\lambda_i$, is considered as a positive example $(\boldsymbol{x},1)$. Similarly, the instance is used to create a negative example $(\boldsymbol{x},0)$ to train $\mathscr{M}_{ij}$ when $\lambda_{\boldsymbol{x}}=\lambda_j$. A base classifier can be implemented by any learning algorithm. Given the outputs of the base classifiers, a decision has to be made for the final classification of the instance. As mentioned in the introduction, we focus on the *weighted voting* (WV) aggregation strategy in order to make this decision. It is an intuitive and very simple strategy which has shown to be very accurate in practice.

Without loss of generality, we assume binary classifiers that map instances to the unit interval [0,1]. In other words, we work with soft classifiers producing *scores* as output, in contrast to discrete classifiers that map to $\{0,1\}$ and thereby produce crisp classifications. The output of a scoring classifier is usually interpreted as a probability or, more generally, as a degree of confidence in the classification. Hence, the WV strategy considers the output $s_{ij}=\mathscr{M}_{ij}(\boldsymbol{x})$ as a weighted "vote" for class label λ_i. Correspondingly, assuming the learners to be additively reciprocal (which is natural in learning by pairwise comparison), we have that[2]

$$s_{ji}\overset{\text{df}}{=}1-s_{ij},$$

and this score is then considered as a weighted vote for class label λ_j. Finally, each class label λ_i is scored in terms of the sum of its votes

$$s_i=\sum_{1\leq j\neq i\leq m}s_{ij}, \tag{1}$$

and the class label with the maximal sum of votes is predicted. Possible ties are often broken at random or are decided in favor of the majority class.

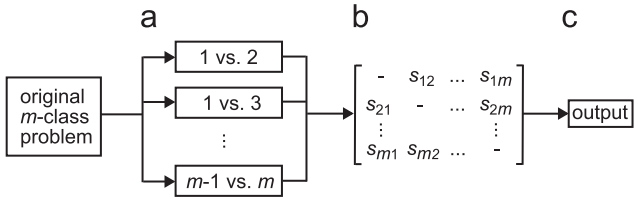

Fig. 1. Basis structure of learning by pairwise comparison: (a) decomposition into binary classification problems, (b) base classifiers provide predictions, and (c) aggregation into one final prediction for the test instance.

3. The setting of label ranking

The optimal voting strategy that we will present is formally analyzed using the label ranking setting. In the next three subsections, respectively, we explain this setting, we show how to use weighted voting to predict a label ranking on the basis of pairwise classifications, and we discuss several benefits of the label ranking setting over the conventional classification setting.

3.1. Generalizing the classification setting

The setting of label ranking can be seen as an extension of the conventional setting of classification [13–15,11]. Roughly speaking, the former is obtained from the latter through replacing single class labels by complete label rankings. So, instead of associating every instance $\boldsymbol{x}$ from an instance space $\mathscr{X}$ with one among a finite set of class labels $\mathscr{L}=\{\lambda_1,\ldots,\lambda_m\}$, we now associate $\boldsymbol{x}$ with a total order of the class labels. This means that we have a complete, transitive, and asymmetric relation $\succ_{\boldsymbol{x}}$ on $\mathscr{L}$ where $\lambda_i\succ_{\boldsymbol{x}}\lambda_j$ indicates that λ_i precedes λ_j in the ranking associated with $\boldsymbol{x}$.

It follows that a ranking can be considered (metaphorically) as a special type of preference relation, and therefore we shall also say that $\lambda_i\succ_{\boldsymbol{x}}\lambda_j$ indicates that λ_i is *preferred* to λ_j given the instance $\boldsymbol{x}$. To illustrate, suppose that instances are students (characterized by attributes such as sex, age, and major subjects in secondary school) and $\succ$ is a preference relation on a fixed set of study fields such as Math, CS, Physics. The goal in label ranking is to learn a "label ranker" in the form of a mapping from the instance space $\mathscr{X}$ to the space of all possible rankings.

Formally, a ranking $\succ_{\boldsymbol{x}}$ can be identified with a permutation $\tau_{\boldsymbol{x}}$ of the set $\{1,\ldots,m\}$. The class of all permutations of this set is denoted by $\mathscr{S}_m$. For ease of presentation, it is convenient to define $\tau_{\boldsymbol{x}}$ such that $\tau_{\boldsymbol{x}}(i)=\tau_{\boldsymbol{x}}(\lambda_i)$ is the position of class label λ_i in

[1] We consider the *strength* of a classifier as its ability to separate classes with high confidence. We will give a precise definition in terms of a parametric probabilistic model in Section 4.1.

[2] In other work, additively reciprocal learners are sometimes called learners that satisfy pairwise consistency; see for example [16].

the ranking. This permutation encodes the following ground truth ranking:

$$\lambda_{\tau_{\boldsymbol{x}}^{-1}(1)} \succ_{\boldsymbol{x}} \lambda_{\tau_{\boldsymbol{x}}^{-1}(2)} \succ_{\boldsymbol{x}} \ldots, \succ_{\boldsymbol{x}} \lambda_{\tau_{\boldsymbol{x}}^{-1}(m)}, \tag{2}$$

where $\tau_{\boldsymbol{x}}^{-1}(j)$ is the index of the class label at position j in the ranking. Clearly, seen the other way around, we have that $\tau_{\boldsymbol{x}}(i) < \tau_{\boldsymbol{x}}(j)$ if and only if $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$. By abuse of terminology, though justified in light of the above one-to-one correspondence, we refer to elements $\tau \in \mathscr{S}_m$ as both permutations and rankings.

In analogy with the conventional classification setting, we do not assume the existence of a deterministic mapping from instances to permutations. Instead, every instance is associated with a *probability distribution* over $\mathscr{S}_m$. This means that there exists a probability distribution $\mathbb{P}(\cdot|\boldsymbol{x})$ for each $\boldsymbol{x} \in \mathscr{X}$ such that, for every $\tau \in \mathscr{S}_m$,

$$\mathbb{P}(\tau|\boldsymbol{x}), \tag{3}$$

is the probability that $\tau_{\boldsymbol{x}} = \tau$ (i.e., for each permutation there is a probability that it is the correct permutation for the instance under consideration). As an illustration, going back to our example, the following probability distribution may be given for a particular student:

Label ranking τ					$\mathbb{P}(\tau\|\boldsymbol{x})$
Math	$\succ$	CS	$\succ$	Physics	0.2526
Math	$\succ$	Physics	$\succ$	CS	0.3789
CS	$\succ$	Math	$\succ$	Physics	0.1684
CS	$\succ$	Physics	$\succ$	Math	0.0421
Physics	$\succ$	Math	$\succ$	CS	0.0947
Physics	$\succ$	CS	$\succ$	Math	0.0632

These probability distributions can be used to reduce a ranking to a single class label prediction. To see this, we note that in the setting of conventional classification, training data consists of tuples $(\boldsymbol{x}_k, \lambda_{\boldsymbol{x}_k})$ which are assumed to be produced according to a probability distribution over $\mathscr{X} \times \mathscr{L}$. This implies that we can associate with each instance $\boldsymbol{x}$ a vector of conditional probabilities

$$p_{\boldsymbol{x}} = (\mathbb{P}(\lambda_1|\boldsymbol{x}), \ldots, \mathbb{P}(\lambda_m|\boldsymbol{x})), \tag{4}$$

where $\mathbb{P}(\lambda_i|\boldsymbol{x})$ denotes the probability of observing the class label $\lambda_{\boldsymbol{x}} = \lambda_i$ given $\boldsymbol{x}$. Now, in label ranking, the class label $\lambda_{\boldsymbol{x}}$ can be naturally associated with the top-label in the ranking $\tau_{\boldsymbol{x}}$, i.e., $\lambda_{\boldsymbol{x}} = \tau_{\boldsymbol{x}}^{-1}(1)$. The probability vector (4) is then the image of the measure in (3) under the mapping $\tau \mapsto \tau^{-1}(1)$. In other words, $\mathbb{P}(\lambda_i|\boldsymbol{x})$ corresponds to the probability that λ_i occurs as a top-label in a ranking τ, and it is computed by summing the probabilities of all possible rankings in which the label is at the first (top) position. In our example, this yields the following probabilities:

$$\mathbb{P}(\text{Math}|\boldsymbol{x}) = 0.6315, \quad \mathbb{P}(\text{CS}|\boldsymbol{x}) = 0.2105, \quad \mathbb{P}(\text{Physics}|\boldsymbol{x}) = 0.1579.$$

To show that the label ranking setting is a proper generalization of the conventional setting of classification, it is also necessary to have a mapping in the reverse direction, i.e., a mapping from probability vectors (4) to measures (3). Such a mapping can be defined in different ways. Since the order of the non-top labels $\mathscr{L} \setminus \{\lambda_{\boldsymbol{x}}\}$ is irrelevant in the classification setting, it appears reasonable to distribute the probability mass $\mathbb{P}(\lambda_i|\boldsymbol{x})$ equally on $\{\tau \in \mathscr{S}_m | \tau^{-1}(1) = \lambda_i\}$. The result is an inverse mapping expressing "indifference" with respect to the order of non-top labels. In our example, this would give the following probability distribution over the rankings:

Label ranking τ					$\mathbb{P}(\tau\|\boldsymbol{x})$
Math	$\succ$	CS	$\succ$	Physics	0.3158
Math	$\succ$	Physics	$\succ$	CS	0.3158
CS	$\succ$	Math	$\succ$	Physics	0.1053
CS	$\succ$	Physics	$\succ$	Math	0.1053
Physics	$\succ$	Math	$\succ$	CS	0.0789
Physics	$\succ$	CS	$\succ$	Math	0.0789

3.2. Predicting a label ranking

Pairwise learning for classification can be extended to the setting of label ranking as follows [17]. It is natural to interpret the output of base classifier $\mathscr{M}_{ij}$ as a decision whether $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ or $\lambda_j \succ_{\boldsymbol{x}} \lambda_i$. Since we are assuming the most general case of scoring classifiers, we again have to "soften" the preference decision. This means that the closer the output of $\mathscr{M}_{ij}$ to 1, the stronger the preference $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ is supported. More formally, we express a soft decision by a valued preference relation $\mathscr{R}_{\boldsymbol{x}}$ for a test instance $\boldsymbol{x}$:

$$\mathscr{R}_{\boldsymbol{x}}(\lambda_i, \lambda_j) = \begin{cases} \mathscr{M}_{ij}(\boldsymbol{x}) & \text{if } i < j, \\ 1 - \mathscr{M}_{ij}(\boldsymbol{x}) & \text{if } i > j. \end{cases} \tag{5}$$

Given a valued preference relation $\mathscr{R}_{\boldsymbol{x}}$ for an instance $\boldsymbol{x}$, the question is how to derive a label ranking from it. This question is non-trivial since such a relation does not always suggest a unique ranking in an unequivocal way [18,19]. Nonetheless, we remark that the weighted voting strategy can be extended to the prediction of a label ranking in a consistent and straightforward way. To this end, class labels are simply ordered according to their total scores, which are again interpreted as individual degrees of support. Thus, each class label λ_i is evaluated by means of the sum of its weighted votes

$$s_i = \sum_{1 \le j \ne i \le m} \mathscr{R}_{\boldsymbol{x}}(\lambda_i, \lambda_j), \tag{6}$$

and a ranking is obtained by ordering according to these evaluations

$$\lambda_i \succ_{\boldsymbol{x}} \lambda_j \Leftrightarrow s_i > s_j. \tag{7}$$

Like for weighted voting in the classification setting, possible ties can be broken at random or decided in favor of the majority class. At first sight, this prediction rule may appear rather ad-hoc. However, in [11] it has been shown that, under mild technical assumptions, (7) is a risk minimizer with respect to the sum of squared rank differences as a loss function on rankings. In other words, (7) maximizes the expected Spearman rank correlation between the predicted ranking and the true ranking.

3.3. Benefits of the label ranking setting

We believe that analyzing the learning by pairwise comparison framework in the label ranking setting is useful for several reasons. Each "vote" of a classifier has a clear and consistent semantics in this setting, namely the degree that λ_i should be ranked higher or lower than λ_j in $\tau_{\boldsymbol{x}}$. More specifically, from a probabilistic perspective, we have the following intuitive interpretation:

$$s_{ij} = \mathbb{P}(\lambda_i \succ_{\boldsymbol{x}} \lambda_j).$$

This offers an interesting alternative to the conventional classification setting in which an output s_{ij} is usually interpreted as the

conditional probability of λ_i given that the class is either λ_i or λ_j [20]:

$$s_{ij} = \mathbb{P}(\lambda_{\boldsymbol{x}} = \lambda_i | \boldsymbol{x}, \lambda_{\boldsymbol{x}} \in \{\lambda_i, \lambda_j\}). \tag{8}$$

Without going into much detail, we mention three reasons why this interpretation is not uncritical, neither semantically nor technically:

- For a new test instance $\boldsymbol{x}$, which is submitted to all base classifiers, the interpretation (8) is actually only valid for those $\mathscr{M}_{ij}$ for which $\lambda_{\boldsymbol{x}} \in \{\lambda_i, \lambda_j\}$ since the probability is conditioned on that event [21]. In other words, a learner $\mathscr{M}_{ij}$ with $\lambda_{\boldsymbol{x}} \notin \{\lambda_i, \lambda_j\}$ is not "competent" for $\boldsymbol{x}$ and, therefore, its prediction is meaningless. In label ranking, this problem does not exist since class label predictions are replaced by preferences and, by definition, each base classifier is "competent" for all instances (since for each instance there exists a total order of the class labels).
- Most machine learning techniques, even those that perform well for classification, do actually yield poor probability estimates [22,23]. This again calls the interpretation (8) into question. Current research on classifier calibration, i.e., transforming classifier outputs into accurate estimations of true conditional probabilities, is still scarce and often classifier-dependent or related to characteristics of the data sets; see for example [24] and references therein.
- Finally, the problem of deriving the probabilities (4) from the pairwise probabilities (8) is non-trivial. It involves $m-1$ variables and $m(m-1)/2$ equality constraints, and thus the system is overconstrained and can only be solved approximately. Different solution methods have different motivations and assumptions, and therefore it is difficult to know which method is most suitable for the data set under consideration [8].

By approaching the aggregation task within the setting of label ranking, we avoid the first problem. Adaptive voting, to be introduced in the next section, addresses also the second issue: instead of requiring a classifier to output probabilities directly, we derive probabilities indirectly by fitting a model to the classifier outputs. Finally, the third problem also does not occur in the label ranking setting since the probability distribution over all rankings (3) can be used to derive the posterior probabilities for all the class labels.

4. Adaptive voting

We introduce adaptive voting (AV) as a novel aggregation strategy for pairwise learning. In Section 4.1, we present its formal framework and in Section 4.2, we show that AV is optimal in the sense that it yields a MAP prediction under certain model assumptions. Finally, in Section 4.3, we discuss the validity of these model assumptions.

4.1. Formal framework

Suppose that, for a particular test instance $\boldsymbol{x}$, the predictions of all base classifiers are given by

$$s(\boldsymbol{x}) \stackrel{\text{df}}{=} \{s_{ij} = \mathscr{M}_{ij}(\boldsymbol{x}) | 1 \leq i \neq j \leq m\}. \tag{9}$$

Adopting a probabilistic perspective, we assume that the output s_{ij} is a random variable. Its distribution depends on whether $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ or $\lambda_j \succ_{\boldsymbol{x}} \lambda_i$; we shall denote the former event by E_{ij} and the latter by E_{ji}. For the classifier $\mathscr{M}_{ij}$ to be accurate, it is natural that values of s_{ij} close to 1 are more probable than values close to 0 when E_{ij} occurs, and vice versa when E_{ji} occurs.

More concretely, we need a model for the probabilities $\mathbb{P}(s_{ij}|\tau)$ of observing scores s_{ij} given a ranking τ. In principle, any type of probability distribution (parametric or non-parametric) can be used for this purpose (compare our discussion of model assumptions in Section 4.3), and indeed, our framework is completely general in this regard. Yet, we shall subsequently assume a particular distribution that has reasonable theoretical properties and is also supported by empirical evidence, namely a truncated exponential distribution

$$\mathbb{P}(s_{ij}|\tau) = c \exp(-\alpha_{ij} \cdot d(s_{ij})), \tag{10}$$

where $c = \alpha_{ij}/(1 - \exp(-\alpha_{ij}))$ is a normalizing constant and α_{ij} is a positive scalar. The term $d(s_{ij})$ is the prediction error defined as $1 - s_{ij}$ when E_{ij} occurs and as s_{ij} when E_{ji} occurs. So, in other words, this term is the distance from the "ideal" outputs 1 and 0, respectively, depending on the event that occurred. Fig. 2 gives an illustration where the score histograms are obtained from two classifiers applied on a representative benchmark data set.

We note that the higher the constant α_{ij} in (10), the more precise the outputs of the classifier $\mathscr{M}_{ij}$ are in the sense of stochastic dominance: For every constant $t \in (0,1)$, the probability to make a small prediction error $d(s_{ij}) \leq t$ increases by increasing α_{ij}. Hence, adapting α_{ij} is a means to take the strength of $\mathscr{M}_{ij}$ into account. This can be done, for example, by using a maximum likelihood estimation, i.e., by maximizing the log-likelihood function

$$\begin{aligned} LL(\alpha_{ij}) &= \sum_{k=1}^{n_{ij}} \log \mathbb{P}(s_{ij}^k | \tau_k) \\ &= n_{ij} \log(\alpha_{ij}) - n_{ij} \log(1 - \exp(-\alpha_{ij})) - \alpha_{ij} \sum_{k=1}^{n_{ij}} d_k, \end{aligned} \tag{11}$$

where n_{ij} is the number of examples $\mathscr{M}_{ij}$ is trained on, $s_{ij}^k = \mathscr{M}_{ij}(\boldsymbol{x}_k)$ is the prediction of the base classifier for the k-th validation instance $\boldsymbol{x}_k$, and $d_k \in [0,1]$ is the corresponding prediction error. To prevent an unwanted bias we advise to use a separate validation set for parameter fitting. Setting the derivative of (11) with respect to α_{ij} equal to zero gives the following implicit solution:

$$\alpha_{ij} = \left(\bar{d} + \frac{1}{\exp(\alpha_{ij}) - 1} \right)^{-1}, \tag{12}$$

where $\bar{d} = \sum_{k=1}^{n_{ij}} d_k / n_{ij}$ is the mean prediction error. This equation cannot be solved explicitly for α_{ij}, but it can be used as an iteration function; see Fig. 3 for a visualization of this function. Thus, starting with an initial value, the value of α_{ij} is updated according to (12), and this is repeated until convergence. A good initial value is $1/\bar{d}$ since the second term in the sum of (12) goes fast to zero when alpha is increased, so the initial value is already close to the solution, and indeed, the whole iteration converges extremely quickly.

For the sake of simplicity and for ease of exposition, we will now assume equal prior probabilities for the events E_{ij} and E_{ji}. Then, the following posterior probabilities $p_{ij} = \mathbb{P}(E_{ij}|s_{ij})$ and $p_{ji} = \mathbb{P}(E_{ji}|s_{ji})$ are obtained by applying Bayes' rule:

$$p_{ij} = \frac{\mathbb{P}(s_{ij}|E_{ij})}{\mathbb{P}(s_{ij})} = \frac{1}{1 + \exp(\alpha_{ij}(1 - 2s_{ij}))}, \tag{13}$$

$$p_{ji} = 1 - p_{ij} = \frac{1}{1 + \exp(-\alpha_{ij}(1 - 2s_{ij}))}. \tag{14}$$

For strong classifiers with a large α_{ij}, these probabilities are "reinforcements" of the original scores s_{ij} in the sense that the p_{ij} are more toward the extreme values of 0 and 1. In contrast, original scores are weakened for classifiers with small α_{ij}, i.e., the scores are pushed toward the indifference value of 0.5. Fig. 4 gives an illustration of these effects. Making an idealized assumption of independence for the base classifiers, the probability $\mathbb{P}(\tau) = \mathbb{P}(\tau|\boldsymbol{x})$ of a ranking $\tau \in \mathscr{S}_m$, given the predictions (9) and corresponding

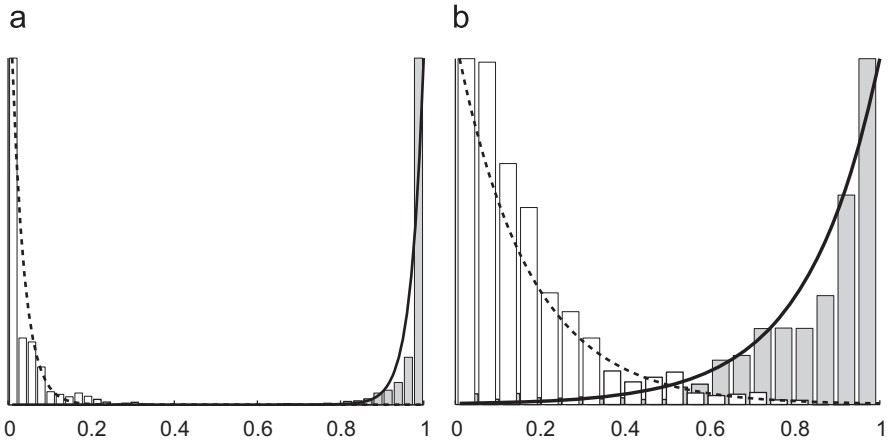

Fig. 2. Two examples of an empirical distribution of s_{ij} (gray bars) and s_{ji} (white bars) together with the estimated exponential distributions. The base classifier depicted in (a) is visibly more certain and accurate in its predictions than the classifier in (b). The height of the bars are scaled to match the estimated distributions.

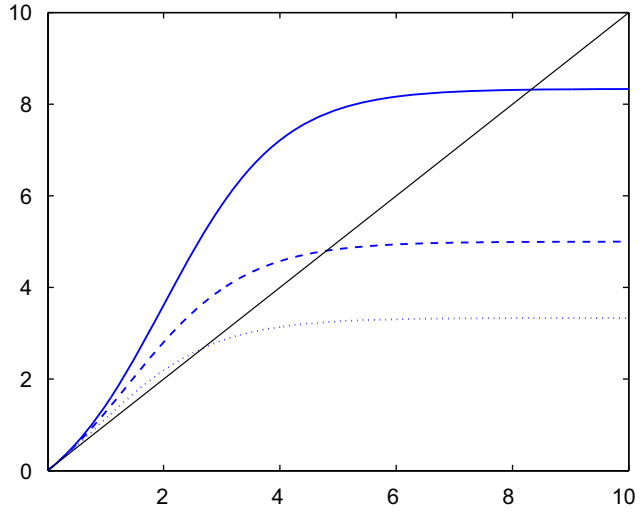

Fig. 3. The iteration procedure as a function of α_{ij} where the mean prediction error is respectively: 0.12 (solid curve), 0.2 (dashed curves), and 0.3 (dotted curve). The diagonal is shown for seeing that the iteration function converges quickly.

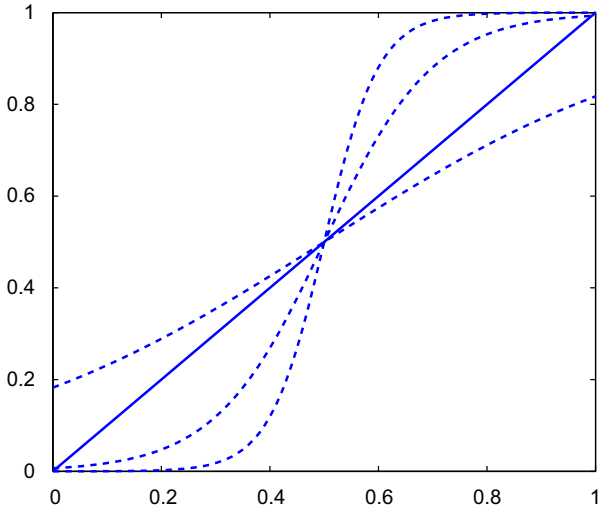

Fig. 4. Original scores s_{ij} (solid curve) and transformations p_{ij} (dashed curves) for α_{ij}-values 1.5, 5, and 10. The higher the value of α_{ij}, the more we approach the step function, representing the strongest reinforcement possible.

posterior probabilities as defined above, is

$$\mathbb{P}(\tau) = c \cdot \prod_{i,j\in\{1,\ldots,m\}:\tau(i)<\tau(j)} p_{i,j}, \tag{15}$$

where c is a normalizing constant. Essentially, this corresponds to a model known as Babington Smith in the statistical literature [25]. The resulting probability distribution over $\mathscr{S}_m$ defined by (15) can serve as a point of departure for various types of inferences. In the following, we shall focus on classification, i.e., estimating the top-label. Needless to say, as the number of different rankings $|\mathscr{S}_m| = m!$ can become very large, it is important to avoid an explicit construction of the distribution over $\mathscr{S}_m$.

4.2. MAP classification

In the conventional classification setting, one is interested in the probabilities $\mathbb{P}(\lambda_k|\boldsymbol{x})$, $\lambda_k \in \mathscr{L}$. Recalling that $\lambda_{\boldsymbol{x}} = \lambda_k$ means that λ_k is the top-label in $\tau_{\boldsymbol{x}}$ (see Section 3), we have the following:

$$\begin{aligned}\mathbb{P}(\lambda_k|\boldsymbol{x}) &= c \sum_{\tau\in\mathscr{S}_m:\tau^{-1}(1)=k} \mathbb{P}(\tau|\boldsymbol{x}) \\ &\propto \prod_{1\le k\neq i\le m} p_{ki} \underbrace{\sum_{\tau\in\mathscr{S}_m:\ \tau^{-1}(1)=k} \prod_{2\le i<j\le m} p_{\tau^{-1}(i),\tau^{-1}(j)}}_{*=1} \\ &= \prod_{1\le k\neq i\le m} p_{ki}. \end{aligned} \tag{16}$$

The expression (*) evaluates to 1 since it is of the form $\sum_{\ell\in\{0,1\}^h} \prod_{i=1}^{h} u_i^{(\ell_i)}$ with $u_i^{(0)} = u_i$, $u_i^{(1)} = 1-u_i$, and by definition $u_i \in [0,1]$.

Scoring the class labels λ_i in terms of the logarithm of their probability gives

$$w_i \stackrel{\text{df}}{=} \log(\mathbb{P}(\lambda_i|\boldsymbol{x})) = \sum_{1\le j\neq i\le m} \log(p_{ij}) = \sum_{1\le j\neq i\le m} w_{ij}, \tag{17}$$

where the individual *adapted scores* are defined as

$$w_{ij} \stackrel{\text{df}}{=} -\log(1+\exp(\alpha_{ij}(1-2s_{ij}))). \tag{18}$$

Hence, the MAP prediction

$$\lambda_{\boldsymbol{x}}^{MAP} = \arg \max_{i=1,\ldots,m} \log(\mathbb{P}(\lambda_i|\boldsymbol{x})), \tag{19}$$

is obtained by finding the class label λ_i for which the sum (17) is maximal. We call this aggregation strategy *adaptive voting* because the original scores s_{ij} are replaced by adapted scores w_{ij} incorporating the strength of the base classifiers $\mathscr{M}_{ij}$. In this way, base classifiers with high performance are seen as more reliable than classifiers with lower performance, and the aggregation strategy becomes less sensitive to (likely incorrect) outputs from the weak and unreliable classifiers.

Since the w_{ij} are non-positive numbers, it may appear more natural to consider their negations $\bar{w}_{ij} \stackrel{\text{df}}{=} -w_{ij}$ as positive "penalties", so that (19) comes down to predicting the class label with lowest total penalty. Indeed, according to (18), a class label λ_i is penalized when s_{ij} is small, and the degree of penalization is in direct correspondence with the probability to observe the output s_{ij} given that λ_i is the true class label: the smaller s_{ij}, the smaller the probability and, hence, the higher the penalty.

The computational complexity of AV is not much higher than that of WV. More specifically, fitting the truncated exponential can be done off-line in time $O(m^2 n)$. At the classification stage, the only difference between AV and WV is the transformation from scores s_{ij} to adapted scores w_{ij}.

4.3. Discussion of model assumptions

Even though is clear that, without any model assumptions, it is impossible to justify a predictor in a theoretical way, let alone to prove its optimality, it is legitimate to ask whether our concrete assumptions are reasonable and realistic. Therefore, we briefly comment on the two main assumptions underlying our adaptive voting method.

The first main assumption is the independence of predictions produced by the base classifiers. Even though it is unlikely to be completely valid in practice, we mention four reasons to defend it. First, independence is routinely assumed in statistics, mainly because without this simplifying assumption, a formal analysis is greatly complicated and often not possible at all. Second, special types of independence assumptions are also made by other successful machine learning methods, notably the famous Naive Bayes classifier. Third, one may argue that assuming any specific type of dependency between the classifiers is at least as speculative as assuming independence. Finally, we note that the assumption of independence is, at least implicitly, also made by weighted voting. In this sense, it is even a necessary prerequisite for our goal to establish a formal connection between adaptive voting and weighted voting (as we shall do in Section 6.1).

The second assumption concerns the (conditional) distribution of scores and is less critical since, as mentioned previously, every type of distribution can in principle be used. Nevertheless, we would like to give some arguments in favor of the truncated exponential distribution (10). Using this model essentially comes down to assuming a monotone behavior, expressing that correct scores are more probable than incorrect ones. This assumption is clearly reasonable for better-than-random classifiers and, moreover, is less restrictive than it may appear at first sight. In fact, the most critical (implicit) property of an exponential model typically concerns the asymptotic behavior (thin tails), which is often violated by real data. Since we use a *truncated* distribution, however, this point is irrelevant in our case. The truncated model is flexible enough to capture a large family of reasonable shapes, ranging from the extreme boundary distribution to the uniform (which, by the way, cannot be obtained with the conventional (non-truncated) exponential). As another advantage, we mention that our model is easily interpretable since the parameter α is in direct correspondence with the strength of a base classifier. Finally, even though it is clear that no distribution will be able to agree with all types of classifiers, it will be shown empirically in Section 7 that (10) is indeed an accurate model for classifiers such as multilayer perceptrons.

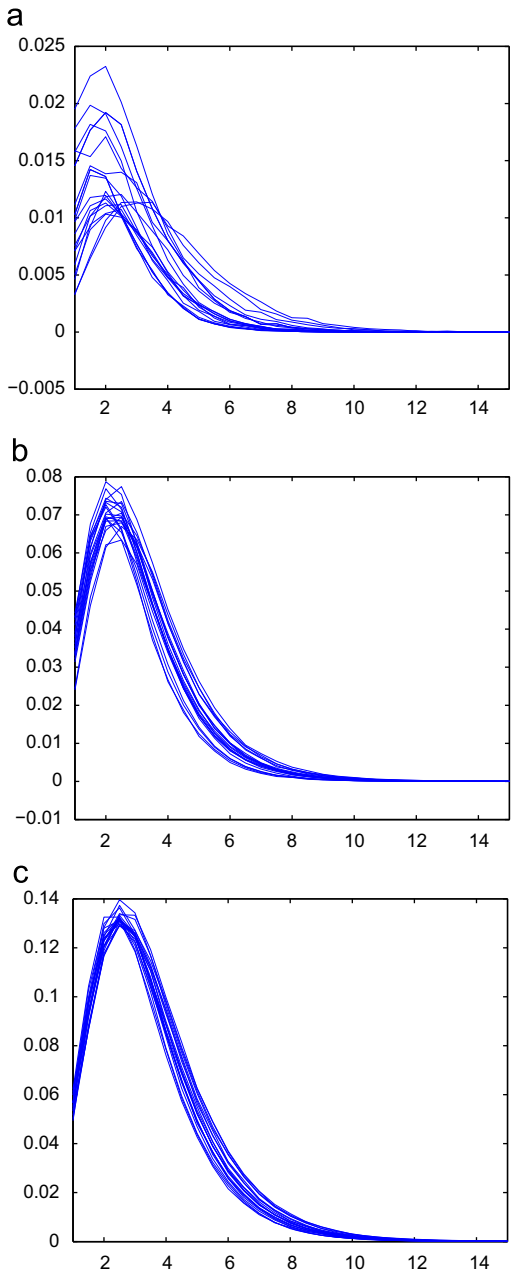

Fig. 5. Expected relative improvement in classification rate comparing AV and WV for: (a) $m = 3$, (b) $m = 6$, and (c) $m = 10$. Relative improvements are shown as a function of the strength of the base classifiers.

5. Simulation studies with synthetic data

In this section, we present two simulation studies to investigate in more detail the effectiveness of AV and its robustness toward estimation errors when fitting the exponential distributions. We compare the classification performance with WV since this aggregation strategy has shown good performance in practice.

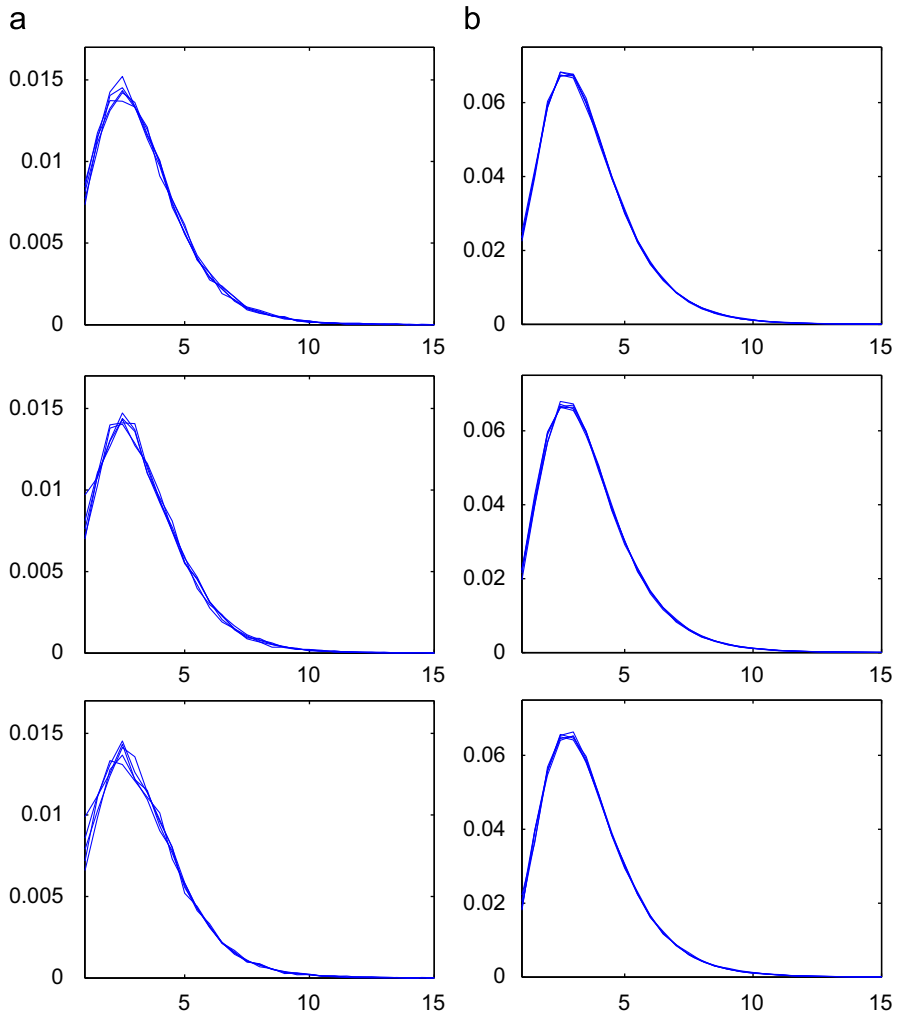

Fig. 6. Expected relative improvement in classification rate comparing AV and WV for: (a) $m=3$, (b) $m=6$ when noise is added to the alphas (top row: 10%, middle row: 20%, and bottom row: 30%). Relative improvements are shown as a function of the strength of the base classifiers.

5.1. Experimental setup and results

The setup of the simulation studies is as follows. We assume that the scores s_{ij} are generated independently according to the truncated exponential distribution (10). For a fixed number of classes m, we generate a random set of α_{ij}-values ($1 \leq i < j \leq m$), each one in the range [1, 3], and then proceed as follows. A random ranking is generated and the output of the base classifiers are computed according to their distributions. These outputs are directly used by WV in order to make a final classification, while AV first adapts them to take the strength of the classifiers into account. The predictions of the aggregation strategies are compared with the top-label in the ranking. This process is repeated for 300.000 times, each time with different rankings and scores. The test statistic to our interest is the expected *relative* improvement $(a-b)/b$, where a is the classification rate of AV and b that of WV. The expectation is computed by averaging the results of the single classifications.

In Fig. 5, we show the expected relative improvement as a function of the strength of the base classifiers. More specifically, each position $t \in [1, 15]$ on the horizontal axis shows the result when the initially generated α_{ij}-values are multiplied with t. The plots are representative for all the initial values that we have generated; in total we generated 200 curves and at least 95% of these curves lie in-between the plotted 20 curves. We may formulate the following three observations. First, AV can improve on WV when its model assumptions are met. Second, there is hardly any difference between the strategies for classifiers with large α_{ij} (so when we focus on the rightmost part of the curves). The reason is that, for very strong base classifiers, the true class label is likely to be a clear winner among all class labels, so an extra adaptation of the scores becomes unnecessary. Third, the benefit of AV over WV increases with the number of classes since there are more scores to adapt (and hereby, more possibilities to correct the final classification of WV). We clearly see this trend when we consider the absolute values on the vertical axis as well as the variance among the curves.

The second simulation study investigates the robustness of AV toward errors in estimating the α_{ij}. The outputs of the base classifiers are computed with the randomly generated α_{ij}, but AV uses a noisy version of them in its calculations. Noise is incorporated by replacing an α_{ij}-value by a randomly drawn number that at most deviates respectively 10%, 20%, and 30% from the true value. Fig. 6 shows the expected relative improvement of AV for each of these three noise levels. The depicted solid curves are again a representative set of all 200 curves that we generated. The curve corresponding to the results obtained by using the true α_{ij} is not depicted since it is

visually almost indistinguishable from the other curves. In general, the true curve can be considered as the highest curve in the figure. For a similar reason, we refrained from presenting results with $m = 10$. We may formulate the following three observations. First, in general, AV is remarkably robust toward estimations errors in the α_{ij}. Second, the robustness increases with the number of classes. Third, independent of the number of classes, inspecting differences in terms of absolute numbers indicates that adding noise has most impact for $\alpha_{ij} < 7$ (the corresponding differences represent more than 95% of all differences). This is again in agreement with our intuition.

5.2. Discussion of the experimental results

In the above experiments, it is not unintentional that the expected improvement is computed with respect to weighted voting, which is known to perform extremely well in practice. Moreover, our experiments with real data sets and classifiers (to be detailed in Section 7) have shown that the predictions of AV and WV are quite likely to coincide, often with a probability significantly higher than 0.9.

To explain this observation, we note that a good probability estimation is a sufficient but not necessary prerequisite for a good classification. More specifically, classification is quite robust toward inaccurate probability estimates: the classification remains correct as long as the highest estimated probability (score) is assigned to the true class label. Consequently, AV and WV will coincide as long as the label with highest w_i does also receive the highest s_i. In fact, there is often a relatively clear winner among the candidate classes, especially if the underlying classification problem is simple or the base classifiers are strong (or both). Note that, as a consequence, all "reasonable" aggregation strategies will perform more or less en par in such cases, so that large differences in performance cannot be expected.

In the next section, the above line of reasoning will be substantiated more formally by showing that WV indeed provides a good approximation of AV.

6. Weighted voting

We now focus on the weighted voting strategy, which has shown excellent performance in practice; see for example [26,12]. We offer a new and formal explanation of this observation by proving that WV yields an approximation to the optimal AV prediction. In addition, we argue that WV can sometimes be considered as being more robust than AV, which is potentially advantageous from a classification point of view.

6.1. Approximate MAP prediction

In this section, it will be shown that weighted voting can be seen as an approximation to adaptive voting, i.e., to the MAP prediction (19). Recalling the posterior probabilities (13) we derive from their logarithm

$$\begin{aligned} w_{ij} &= \log(\mathbb{P}(s_{ij}|E_{ij})) - \log(\mathbb{P}(s_{ij}|E_{ij}) + \mathbb{P}(s_{ij}|E_{ji})) \\ &= \alpha_{ij} \cdot s_{ij} - \alpha_{ij} + e(d_{ij}, \alpha_{ij}), \end{aligned} \tag{20}$$

where

$$e(d_{ij}, \alpha_{ij}) \stackrel{\text{df}}{=} -\log(\exp(-\alpha_{ij} d_{ij}) + \exp(-\alpha_{ij}(1 - d_{ij}))). \tag{21}$$

This term, which depends on α_{ij} and the prediction error $d_{ij} = d(s_{ij})$ of classifier $\mathscr{M}_{ij}$, is bounded in size by $|\alpha_{ij}/2 - \log(2)|$. Since $\exp(x) \approx 1 + x$ for small x, we have that $e(d_{ij}, \alpha_{ij})$ is small and nearly constant for small α_{ij}. For larger α_{ij}, we have $e(d_{ij}, \alpha_{ij}) \approx \log(1 + e^{-\alpha_{ij}}) \approx 0$ at least for d_{ij} close to 1 or close to 0, so that

$$w_{ij} \approx \alpha_{ij}(s_{ij} - 1). \tag{22}$$

In summary, we can conclude that

$$w_i \approx \sum_{1 \le j \ne i \le m} \alpha_{ij} \cdot s_{ij} - \sum_{1 \le j \ne i \le m} \alpha_{ij},$$

if the α_{ij} are either not too large or if the predictions are precise, which in turn is likely in case of large α_{ij}. We note that this is in perfect agreement with the results of our simulation studies in which the maximal differences between AV and WV have been observed for alphas of medium size.

If we furthermore assume that the strengths of the classifiers are not too different, that is, $\alpha_{ij} \approx \alpha$ for all $1 \le i < j \le m$, then we have

$$w_i \approx \alpha \sum_{1 \le j \ne i \le m} s_{ij} + \text{const} = \alpha \cdot s_i + \text{const}. \tag{23}$$

In other words, the scores obtained by weighted voting yield an approximate affine transformation of the theoretically optimal score $\log(\mathbb{P}(\lambda_i))$ and, hence, are likely to produce the same or a similar ordering of the class labels λ_i, $i = 1, \ldots, m$. We may conclude that, under the above assumptions, weighted voting provides a good approximation of the MAP prediction (19).

Nevertheless, the above derivation has also shown that AV and WV may not coincide in cases where the α_{ij} are rather different and, moreover, when some base classifiers produce poor estimates. Thus, it is still possible that AV will produce better results in practice, and in fact, this hope is supported by our simulation results in Section 5. However, recalling that these results have been obtained under idealized conditions in which the model assumptions underlying AV are completely valid, one may also suspect that AV could fail if these assumptions are not satisfied. And indeed, apart from its approximation properties, WV seems to have the advantage of making less assumptions and, therefore, being more robust toward deviations from expected score distributions. Before presenting experimental results in the next section, we elaborate on this aspect in more detail.

6.2. Robustness toward inaccurate scores

The scores (18) are rescalings of the original scores $s_{ij} \in [0, 1]$ to a potentially larger range $[-\log(1 + \exp(\alpha_{ij})), -\log(1 + \exp(-\alpha_{ij}))] \subset (-\infty, 0]$. It follows that the sum of these adapted scores, w_i, may have a considerably higher variance than the corresponding s_i. Moreover, it can happen that a class is disqualified by a single small adapted score w_{ij} or, equivalently, by a large penalty $\bar{w}_{ij}$. While being correct and fully legitimate from a theoretical point of view, the penalization in AV may become problematic if its model assumptions are violated. In particular, recalling that the $\bar{w}_{ij}$ are in direct correspondence with the probabilities to observe a score s_{ij} given that λ_i is the true class, it becomes obvious that an "over-penalization" will occur in cases where the probability of a small s_{ij} is under-estimated by the exponential model underlying AV.

As an illustration, inspired by our experiments with decision trees (see the next section), consider the fit of the distribution shown in Fig. 7. Since this distribution is discrete and far from being exponential, the fit is obviously poor. More importantly, the probability of a small score 0.05 is strongly underestimated. Consequently, if this score is output (which happens with probability 0.2), the class label will be strongly punished and is unlikely to win the adapted voting procedure.

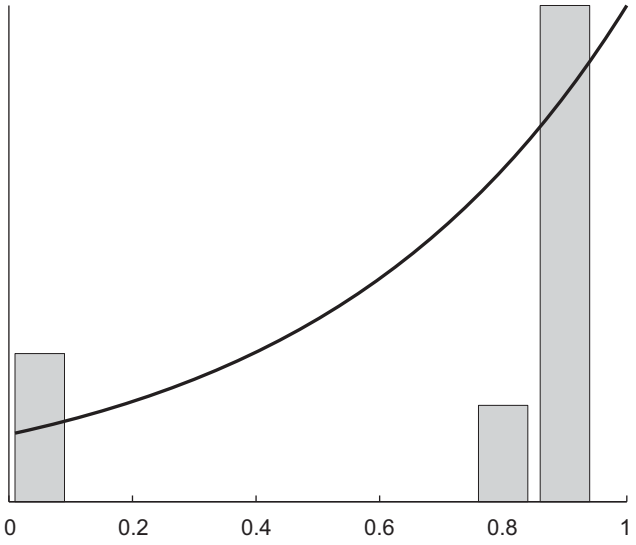

Fig. 7. A score distribution with three possible values (0.05, 0.8, and 0.9) for $\lambda_i \succ \lambda_j$. The corresponding probabilities are 0.20, 0.13, and 0.67. The maximum likelihood fit gives a value of $\alpha_{ij} \approx 1.3947$.

More concretely, consider a scenario with $m=4$ classes, and suppose that all base classifiers have the distribution shown in Fig. 7. Moreover, suppose that λ_1 is the true class label, and that the following scores are produced:

$$[s_{ij}]_{i\neq j} = \begin{bmatrix} - & 0.90 & 0.90 & 0.05 \\ 0.10 & - & 0.90 & 0.80 \\ 0.10 & 0.10 & - & 0.80 \\ 0.95 & 0.20 & 0.20 & - \end{bmatrix}.$$

Obviously, the learner $\mathscr{M}_{14}$ has made an incorrect prediction. WV tolerates this error in the sense that it still assigns the highest score (namely 1.85) to λ_1. According to AV, however, class label λ_2 is better than λ_1: the latter is penalized by ≈ 2.07, while the former has a smaller penalty of ≈ 2.04. As explained above, the main reason is that the small score of 0.05 produced by $\mathscr{M}_{14}$ is overpenalized.

We can see the above effect also from the other way around, that is, the scores s_{ij} used in weighted voting can be considered as "regularized" scores which are normalized to the range [0, 1], thereby especially reducing the effect of small scores. While perhaps being suboptimal for probability estimation, this can be reasonable from a classification point of view: As long as the true class receives correct votes, i.e., high scores, nothing will change anyway. However, if the true class receives one (or even more) incorrect votes, an aggregation strategy which is tolerant toward low scores is more likely to preserve this class as the overall winner than a strategy which is more sensitive in this regard.

Interestingly, the situation is to some extent comparable to estimating conditional probabilities of attributes given classes, $\mathbb{P}(a|\lambda_i)$, in Naive Bayes classification. Estimating these probabilities by relative frequencies yields unbiased estimates, but it causes the problem that small probabilities have an extreme influence on the rank position of a class. In particular, if a single probability is 0 (since the attribute value of a has not yet been observed for λ_i), then multiplication of all conditional probabilities causes the probability of the class label to become 0 as well. In practice, probabilities are therefore estimated by using a Laplace correction or another smoothing technique [27]. A similar problem occurs in AV so that, as soon as one of the probabilities p_{ki} in (16) becomes small, the probability of the true class label λ_k becomes small as well. Correcting transformed scores in AV is not a trivial task since scores lie in different intervals (unbounded from one side).

7. Experimental analysis

In this section, we provide an extensive empirical evaluation and comparison between WV and AV using benchmark data sets. We also include binary voting (BV) for two reasons. First, it is often used in practice as an alternative to WV. Second, this strategy maps the original scores s_{ij} to transformations $b_{ij} \in \{0,1\}$ where $b_{ij}=1$ iff $s_{ij} \geq 0.5$. So, BV can also be seen as a reinforcement of the outputs of the base classifiers, although this happens independently of their strength (in Fig. 4, this reinforcement would correspond to the step function).

7.1. Data sets and base classifiers

To compare BV, WV, and AV, experiments have been conducted on a collection of 17 benchmark data sets obtained from the UCI machine learning repository and the StatLib project [28,29]. We have chosen these data sets because they vary greatly in size, number of classes, and other characteristics such as class distribution. We performed experiments with several learning algorithms to produce base classifiers. Below we present three of them. All learning algorithms produce multi-class classifiers, but it has been shown that even these classifiers benefit from a pairwise decomposition [7]. In addition, and more importantly, we have chosen these algorithms because they are representative in the sense that they learn base classifiers that satisfy or do not satisfy to a certain degree the AV assumptions.

Our first classifier is a multilayer perceptron (MLP) where each network node has a sigmoid transfer function. As we could verify by statistical goodness-of-fit tests, the scores produced by MLPs are sufficiently well in agreement with our model assumptions. That is, the scores s_{ij} produced by a classifier $\mathscr{M}_{ij}$ can usually be fitted by an exponential model (10); see again Fig. 2. It is our hypothesis that AV can outperform WV and BV for this classifier. We make the same claim for our second classifier, which is a distance-weighted k-nearest neighbor classifier (k-NN), although admittedly, the exponential model is this time less clearly pronounced. This classifier computes scores $s_{ij} = (\sum_{l=1}^{k} e_l/d_l)/(\sum_{l=1}^{k} 1/d_l)$, where d_l is the distance of the l-th neighbor; moreover, $e_l = 1$ if this neighbor is from class λ_i and $e_l = 0$ if its class is λ_j. Just for comparison, and also to investigate the robustness of adaptive voting toward strong violations of its model assumptions, we included a third base classifier: J48, an implementation of the C4.5 decision tree learner. This classifier outputs relative class frequencies in the leaves as scores. These scores are relatively poor and hard to fit by an exponential model; see Fig. 8. In particular, the scores do often not exhaust the whole interval [0, 1] but instead only produce a limited number of different values. This makes a good estimation of the α_{ij} difficult. Therefore, we hypothesize that in this case AV will not yield a gain in performance.

7.2. Experimental setup

In our experiments, we used the WEKA machine learning software [30]. The experimental setup was as follows. For all three base learners we used the default options, except for a variable learning rate in MLP, a fixed number of 10 nearest neighbors in k-NN, and Laplace correction for probability estimation in J48. Stratified tenfold cross validation is applied and repeated for five times, each time with a different random permutation of the data set. The alpha values for AV are obtained by maximum likelihood estimation on an

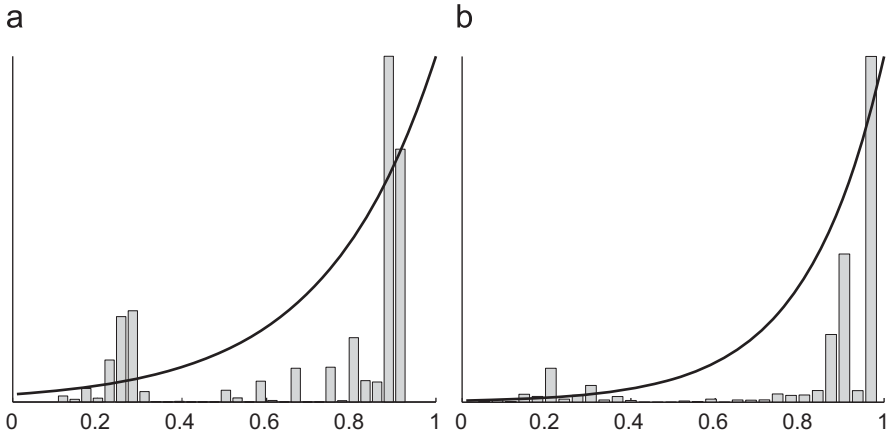

Fig. 8. Two examples of an empirical distribution of the scores s_{ij} for $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ together with the estimated exponential distribution when using J48 as base classifier. The height of the bars are scaled to match the estimated distributions.

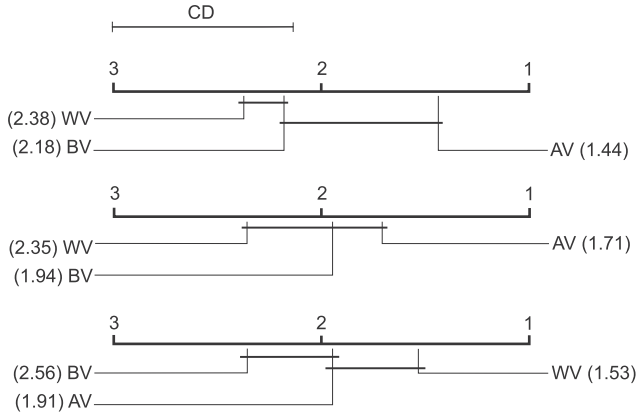

Fig. 9. Comparison of aggregation strategies on the basis of the Nemenyi test, using the error rate as performance metric: (top) MLP, (middle) k-NN, and (below) J48.

independent 20% subset of the training data. Ties among the ranking of class labels occurred for binary voting, but only sporadic on some data sets. We simply resolved these ties at random.

7.3. Experimental results

In a first experiment, we compute error rates as averages over the cross-validation runs. For readability and completeness, we refrained from presenting error rates for all aggregation strategies and base classifiers in the main text; instead, these test statistics are gathered in Appendix A in Tables A1–A3. We summarized these values as pairwise win-loss-equal statistics and with critical distance (CD) diagrams. Since both evaluation methods agreed on the conclusions that we can draw from them, we will only present the CD diagrams since they are easier to understand; see Fig. 9. These diagrams depict the result of the Nemenyi test which has been advocated as stronger than other widely-used significance tests [31]. The test compares the average ranks of the strategies over all data sets. A lower average rank implies a better aggregation strategy (thus, strategies on the right side in the diagrams are better) and strategies that are not significantly different when compared to each other are connected through a bold line. The significance level is $\alpha = 0.05$ which implies that two strategies are significantly different when the difference between their average ranks is at least $CD = 0.88$.

The results confirm what could be expected from our theoretical considerations and the simulation results. More specifically, for MLPs we have a significant improvement by AV when compared to WV. Also, from the fact that BV is ranked better than WV, we can conclude that reinforcements are beneficial, yet the strength of the classifiers have to be taken into account. For the nearest neighbor classifier we have identical observations and conclusions although we cannot guarantee that the differences are statistically significant. For J48, the classifier that strongly violates our model assumptions, we see that WV wins significantly from BV, but not from AV which this time is ranked second. As a side note, we also tried to use unpruned decision trees with the idea that increasing the number of possible scores (making the score distribution less discrete) could give rise to a better fit of the exponential model. However, from statistical tests, we did not see a significant gain when using unpruned trees.

As a final experiment, we are interested in the robustness of the strategies with respect to inaccurate scores of the base classifiers. To investigate this issue, we note that the final classifications of the aggregation strategies often coincide; see Table C1 for details. This implies that the strategies often make mistakes for the same instances, and a measure for how severe the mistake is will give a good indication about the robustness of the strategy toward inaccurate votes. For this reason, we apply experiments in the same setting as above but replace the error rate as a loss function by the normalized position error which is defined by

$$\frac{\tau^{-1}(\lambda_{\boldsymbol{x}}) - 1}{m - 1} \in \{0, 1/(m-1), \ldots, 1\},$$

where we recall that $\tau^{-1}(\lambda_{\boldsymbol{x}})$ is the position of the true class label $\lambda_{\boldsymbol{x}}$ in the predicted ranking τ and m the number of classes [32]. Hence, the larger the normalized position error, the further away the true class label is from the top position in the predicted ranking. Values of this performance metric are gathered in Appendix B in Tables B1–B3 and corresponding CD diagrams are presented in Fig. 10.

Interestingly, the ranking of the aggregation strategies is indeed sometimes different compared to that when we consider error rate. For MLP we still have the same ranking and for k-NN we have that AV changed second place with BV, but differences are not significant. The ranking for J48 shows that AV makes the largest incorrect

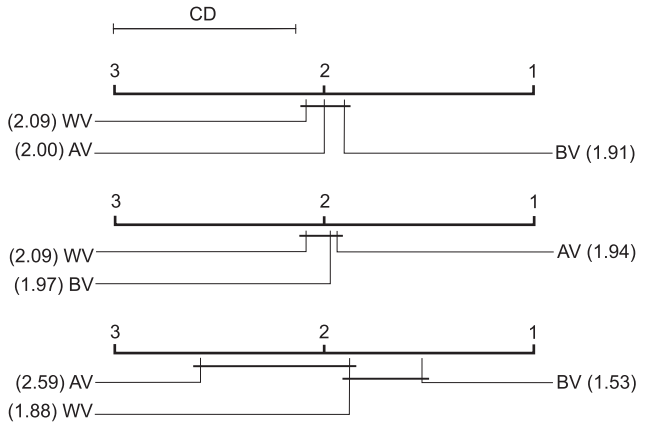

Fig. 10. Comparison of aggregation strategies on the basis of the Nemenyi test, using the normalized position error as performance metric: (top) MLP, (middle) *k*-NN, and (below) J48.

predictions in the sense that, when an incorrect prediction is made, the true class label is not at all among the labels that try to compete for the top-label in the predicted ranking. So, large variance among the scores in combination with inaccurate votes is disadvantageous for adaptive voting. Weighted voting and binary voting are clearly more robust. All these results are again in correspondence with our previous theoretical analysis, simulation studies, and discussions.

In light of the above results, we may also explain why the binary voting strategy performs quite reasonable in terms of error rate and normalized position error, despite its apparent simplicity. In a sense, BV can be seen as a combination of AV and WV. Like AV, it reinforces scores s_{ij}, albeit in a fixed rather than adaptive manner (always mapping to 0 or 1). Thus, it considers all classifiers as perfect, an assumption which is only approximately true for an ensemble of strong learners. Like WV, however, the scores remain in the unit interval and thereby BV is less sensitive toward inaccurate probability estimates than AV (recall the discussion in Section 6.2).

7.4. Beyond the truncated exponential distribution

Throughout the paper, we advocated the truncated exponential distribution for fitting scores, in particular as it ensures monotonicity while still being quite flexible. Yet, an obvious question is whether, in some cases, a more flexible model could be advantageous. To address this question, we conducted additional experiments with two other types of distributions, namely the beta distribution and kernel density estimation. In the following, we briefly summarize our main findings, without going into technical details.

The beta distribution is a family of continuous probability distributions on the unit interval, well-known in the field of statistics [33]. Depending on its two scalar parameters, the distribution can be a U-shaped curve, straight line, strictly convex, monotone increasing or decreasing, and so on. Nevertheless, for MLP, AV with the exponential is still the best aggregation strategy. As expected, AV with beta distribution comes close, since the beta distribution is often shaped as an exponential. For J48, AV performs much better with a beta distribution, but it is still worse than WV (albeit this time there is no statistically significant difference). For the third base learner, weighted *k*-NN, we obtained poor results despite several efforts to improve the fits. We found that, in this case, the beta distribution is too flexible and often gives non-monotone distributions that severely over- or underestimate the probability of unseen scores (which occur frequently for this classifier).

The application of a kernel density estimator was more difficult, especially as the results turned out to be extremely sensitive toward the choice of the bandwidth. Moreover, even a careful tuning of this critical parameter could not avoid that, in some cases, quite non-intuitive (e.g., non-monotone) density functions were produced as estimates. And indeed, for MLP and *k*-NN, we were unable to outperform the results obtained with the truncated exponential.

In summary, we may conclude that more flexibility is not necessarily an advantage. On the one hand, it is true that more flexible models can in principle yield better approximations. On the other hand, fitting such models is also more difficult and more likely to fail. Roughly speaking, a good fit of a simple model, even if it is slightly too restrictive, is often better than a poor fit of a more flexible model which is difficult to estimate.

8. Conclusions and outlook

In this paper, we have studied the problem of aggregating predictions in pairwise classification, a special binary decomposition technique commonly used in practice. In this regard, two important contributions have been made. First, the method of adaptive voting has been derived in the formal framework of label ranking. Adaptive voting is a generalized voting strategy in which the predictions of base classifiers are adapted according to their strength. Under our model assumptions, it is provably optimal in the sense of yielding a MAP prediction of the class label of a test instance. Second, we offered hitherto missing theoretical arguments in favor of weighted voting as a quasi-optimal aggregation strategy in pairwise classification and, thereby, improve the understanding of its good performance in practice. Roughly speaking, weighted voting approximates the optimal adaptive voting prediction. Moreover, compared with adaptive voting, it has the additional advantage of being more robust in situations where the AV model assumptions are violated. Our empirical results are in perfect agreement with all theoretical considerations. In summary, we have shown that weighted voting is quite competitive, even though slight but consistent improvements can be achieved by adaptive voting, provided its underlying model assumptions are approximately valid.

The formal framework that we have used for our analysis of aggregation strategies is interesting in its own right and may provide the basis for further developments. Indeed, one should note that there is space for further improving adaptive voting, namely by relaxing some model assumptions mainly needed to adhere to corresponding properties of weighted voting. For example, one could think of incorporating prior probabilities in the estimation of the conditional class probabilities p_{ij}, and using asymmetric distributions to model the scores produced by a classifier $\mathscr{M}_{ij}$ (i.e., the distribution of scores given that $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ is not necessarily the same, except for reflection, as the distribution of scores given that $\lambda_j \succ_{\boldsymbol{x}} \lambda_i$). Also, the maximum likelihood approach for estimating the strengths of base classifiers is susceptible to over-fitting, so using other estimation techniques might be advisable.

We finally note that our framework of label ranking is quite general and not restricted to the conventional classification problem. Instead, it also allows for studying other problems including multi-label classification and, of course, label ranking itself. Problems of that kind shall be addressed in future work.

Appendix A. Detailed results for error rate

The error rate for BV, WV, and AV on the 17 data sets using multilayer perceptron, distance-weighted *k*-nearest neighbor and J48 as the base classifiers is presented in Tables A1–A3.

Table A1
The error rate for BV, WV, and AV on the 17 data sets using multilayer perceptron as the base classifier.

Data set	BV	WV	AV
Analcat-author	0.0029 ± 0.0057	0.0033 ± 0.0064	0.0033 ± 0.0059
Balance-scale	0.1412 ± 0.0296	0.1403 ± 0.0306	0.1409 ± 0.0304
Cars	0.2950 ± 0.0664	0.2930 ± 0.0596	0.2920 ± 0.0603
CMC	0.4719 ± 0.0391	0.4717 ± 0.0405	0.4706 ± 0.0399
Eucalyptus	0.3394 ± 0.0703	0.3348 ± 0.0674	0.3344 ± 0.0637
Glass	0.5121 ± 0.0620	0.5345 ± 0.0571	0.5279 ± 0.0592
MFEAT-Fourier	0.1582 ± 0.0258	0.1626 ± 0.0253	0.1585 ± 0.0270
MFEAT-Karhunen	0.0370 ± 0.0133	0.0364 ± 0.0138	0.0363 ± 0.0141
MFEAT-Morpho	0.2853 ± 0.0250	0.2841 ± 0.0264	0.2837 ± 0.0257
MFEAT-Zernike	0.1683 ± 0.0210	0.1747 ± 0.0216	0.1673 ± 0.0204
Optdigits	0.0168 ± 0.0047	0.0177 ± 0.0048	0.0164 ± 0.0047
Pendigits	0.0187 ± 0.0041	0.0192 ± 0.0041	0.0174 ± 0.0038
Page-blocks	0.0509 ± 0.0066	0.0628 ± 0.0066	0.0494 ± 0.0071
Segment	0.0551 ± 0.0148	0.0562 ± 0.0158	0.0552 ± 0.0147
Vehicle	0.2643 ± 0.0443	0.2621 ± 0.0442	0.2618 ± 0.0432
Vowel	0.1984 ± 0.0347	0.2491 ± 0.0418	0.2105 ± 0.0416
Waveform	0.1374 ± 0.0143	0.1371 ± 0.0144	0.1372 ± 0.0144

Reported values are the averages over the test folds followed by the standard deviation.

Table A2
The error rate for BV, WV, and AV on the 17 data sets using distance-weighted k-nearest neighbor as the base classifier.

Data set	BV	WV	AV
Analcat-author	0.0036 ± 0.0073	0.0036 ± 0.0073	0.0036 ± 0.0073
Balance-scale	0.1077 ± 0.0284	0.1064 ± 0.0280	0.1056 ± 0.0268
Cars	0.2434 ± 0.0652	0.2484 ± 0.0633	0.2434 ± 0.0652
CMC	0.5257 ± 0.0341	0.5255 ± 0.0361	0.5243 ± 0.0363
Eucalyptus	0.4518 ± 0.0548	0.4556 ± 0.0568	0.4531 ± 0.0546
Glass	0.3268 ± 0.0815	0.3349 ± 0.0813	0.3268 ± 0.0815
MFEAT-Fourier	0.1833 ± 0.0266	0.1831 ± 0.0240	0.1833 ± 0.0266
MFEAT-Karhunen	0.0384 ± 0.0121	0.0391 ± 0.0126	0.0384 ± 0.0121
MFEAT-Morpho	0.2892 ± 0.0257	0.2887 ± 0.0253	0.2890 ± 0.0256
MFEAT-Zernike	0.1902 ± 0.0195	0.1904 ± 0.0198	0.1902 ± 0.0195
Optdigits	0.0128 ± 0.0047	0.0131 ± 0.0046	0.0128 ± 0.0047
Pendigits	0.0073 ± 0.0024	0.0074 ± 0.0024	0.0073 ± 0.0024
Page-blocks	0.0419 ± 0.0080	0.0422 ± 0.0081	0.0419 ± 0.0080
Segment	0.0476 ± 0.0117	0.0475 ± 0.0121	0.0476 ± 0.0117
Vehicle	0.2869 ± 0.0414	0.2893 ± 0.0438	0.2869 ± 0.0414
Vowel	0.0729 ± 0.0282	0.3974 ± 0.0583	0.0729 ± 0.0282
Waveform	0.2063 ± 0.0166	0.2058 ± 0.0165	0.2063 ± 0.0166

Reported values are the averages over the test folds followed by the standard deviation.

Table A3
The error rate for BV, WV, and AV on the 17 data sets using J48 as the base classifier.

Data set	BV	WV	AV
Analcat-author	0.0471 ± 0.0231	0.0440 ± 0.0230	0.0492 ± 0.0243
Balance-scale	0.2063 ± 0.0384	0.2066 ± 0.0398	0.2070 ± 0.0401
Cars	0.1641 ± 0.0621	0.1664 ± 0.0588	0.1689 ± 0.0620
CMC	0.4715 ± 0.0424	0.4696 ± 0.0415	0.4694 ± 0.0415
Eucalyptus	0.3781 ± 0.0515	0.3742 ± 0.0553	0.3736 ± 0.0554
Glass	0.2949 ± 0.0879	0.2763 ± 0.0854	0.2744 ± 0.0879
MFEAT-Fourier	0.2165 ± 0.0256	0.2058 ± 0.0264	0.2087 ± 0.0249
MFEAT-Karhunen	0.1170 ± 0.0240	0.1060 ± 0.0236	0.1079 ± 0.0247
MFEAT-Morpho	0.2748 ± 0.0208	0.2731 ± 0.0224	0.2731 ± 0.0219
MFEAT-Zernike	0.2430 ± 0.0238	0.2326 ± 0.0252	0.2330 ± 0.0259
Optdigits	0.0578 ± 0.0099	0.0513 ± 0.0099	0.0527 ± 0.0108
Pendigits	0.0302 ± 0.0050	0.0285 ± 0.0049	0.0293 ± 0.0053
Page-blocks	0.0293 ± 0.0060	0.0290 ± 0.0058	0.0294 ± 0.0062
Segment	0.0320 ± 0.0117	0.0320 ± 0.0108	0.0318 ± 0.0103
Vehicle	0.2880 ± 0.0605	0.2801 ± 0.0577	0.2858 ± 0.0549
Vowel	0.1966 ± 0.0359	0.1994 ± 0.0410	0.1964 ± 0.0378
Waveform	0.2371 ± 0.0221	0.2323 ± 0.0224	0.2331 ± 0.0227

Reported values are the averages over the test folds followed by the standard deviation.

Appendix B. Detailed results for normalized position error

The normalized position error for BV, WV, and AV on the 17 data sets using multilayer perceptron, distance-weighted k-nearest neighbor and J48 as the base classifiers is presented in Tables B1–B3.

Table B1
The normalized position error for BV, WV, and AV on the 17 data sets using multilayer perceptron as the base classifier.

Data set	BV	WV	AV
Analcat-author	0.2974 ± 0.1496	0.2976 ± 0.1506	0.3054 ± 0.1546
Balance-scale	0.3463 ± 0.1586	0.3869 ± 0.1878	0.4743 ± 0.2412
Cars	0.3870 ± 0.0996	0.3825 ± 0.1042	0.3782 ± 0.1079
CMC	0.4373 ± 0.0789	0.4364 ± 0.0781	0.4399 ± 0.0791
Eucalyptus	0.4055 ± 0.1020	0.4062 ± 0.1051	0.4120 ± 0.1113
Glass	0.2530 ± 0.0324	0.2549 ± 0.0337	0.2858 ± 0.0438
MFEAT-Fourier	0.4062 ± 0.2098	0.4082 ± 0.2093	0.4076 ± 0.2097
MFEAT-Karhunen	0.3758 ± 0.1937	0.3735 ± 0.1920	0.3726 ± 0.1918
MFEAT-Morpho	0.4095 ± 0.1953	0.4144 ± 0.1986	0.4079 ± 0.1948
MFEAT-Zernike	0.3968 ± 0.2034	0.4005 ± 0.2033	0.3999 ± 0.2051
Optdigits	0.5045 ± 0.1172	0.5064 ± 0.1179	0.5024 ± 0.1173
Pendigits	0.3530 ± 0.1864	0.3514 ± 0.1855	0.3518 ± 0.1869
Page-blocks	0.0385 ± 0.0095	0.0384 ± 0.0081	0.0372 ± 0.0079
Segment	0.4808 ± 0.2457	0.4792 ± 0.2438	0.4794 ± 0.2412
Vehicle	0.4479 ± 0.1463	0.4452 ± 0.1416	0.4424 ± 0.1487
Vowel	0.4130 ± 0.1997	0.4163 ± 0.1968	0.4153 ± 0.2025
Waveform	0.4590 ± 0.2125	0.4590 ± 0.2126	0.4594 ± 0.2139

Reported values are the averages over the test folds followed by the standard deviation.

Table B2
The normalized position error for BV, WV, and AV on the 17 data sets using distance-weighted k-nearest neighbor as the base classifier.

Data set	BV	WV	AV
Analcat-author	0.3033 ± 0.1534	0.3025 ± 0.1530	0.2908 ± 0.1460
Balance-scale	0.3964 ± 0.1965	0.3994 ± 0.2011	0.4614 ± 0.2475
Cars	0.3803 ± 0.1073	0.3801 ± 0.1064	0.3816 ± 0.1158
CMC	0.4467 ± 0.0675	0.4472 ± 0.0682	0.4459 ± 0.0672
Eucalyptus	0.4155 ± 0.0909	0.4142 ± 0.0927	0.4197 ± 0.1023
Glass	0.3423 ± 0.0719	0.3388 ± 0.0738	0.3524 ± 0.0883
MFEAT-Fourier	0.4206 ± 0.2159	0.4197 ± 0.2156	0.4006 ± 0.2034
MFEAT-Karhunen	0.3715 ± 0.1975	0.3719 ± 0.1978	0.3793 ± 0.2000
MFEAT-Morpho	0.4106 ± 0.1975	0.4077 ± 0.1959	0.4011 ± 0.1940
MFEAT-Zernike	0.4143 ± 0.2118	0.4145 ± 0.2122	0.4158 ± 0.2127
Optdigits	0.4146 ± 0.2068	0.4155 ± 0.2072	0.4513 ± 0.2280
Pendigits	0.3527 ± 0.1861	0.3537 ± 0.1867	0.3587 ± 0.1934
Page-blocks	0.0415 ± 0.0121	0.0411 ± 0.0120	0.0391 ± 0.0099
Segment	0.4713 ± 0.2397	0.4726 ± 0.2406	0.4431 ± 0.2351
Vehicle	0.4667 ± 0.1439	0.4674 ± 0.1440	0.4560 ± 0.1420
Vowel	0.4001 ± 0.1986	0.4045 ± 0.1787	0.3951 ± 0.1998
Waveform	0.4651 ± 0.1961	0.4651 ± 0.1960	0.4650 ± 0.1967

Reported values are the averages over the test folds followed by the standard deviation.

Table B3
The normalized position error for BV, WV, and AV on the 17 data sets using J48 as the base classifier.

Data set	BV	WV	AV
Analcat-author	0.3288 ± 0.1564	0.3295 ± 0.1604	0.3364 ± 0.1635
Balance-scale	0.3116 ± 0.1105	0.3124 ± 0.1102	0.4762 ± 0.2049
Cars	0.3869 ± 0.1403	0.3956 ± 0.1464	0.4077 ± 0.1571
CMC	0.4285 ± 0.0825	0.4322 ± 0.0820	0.4301 ± 0.0841
Eucalyptus	0.4075 ± 0.0993	0.4050 ± 0.0998	0.4139 ± 0.1162
Glass	0.3501 ± 0.0752	0.3481 ± 0.0771	0.3564 ± 0.0857
MFEAT-Fourier	0.4204 ± 0.2057	0.4227 ± 0.2057	0.4252 ± 0.2077
MFEAT-Karhunen	0.3816 ± 0.1864	0.3790 ± 0.1861	0.3865 ± 0.1879
MFEAT-Morpho	0.3910 ± 0.1909	0.3942 ± 0.1929	0.3970 ± 0.1906
MFEAT-Zernike	0.4175 ± 0.2021	0.4188 ± 0.2032	0.4173 ± 0.2010
Optdigits	0.4125 ± 0.2029	0.4128 ± 0.2029	0.4243 ± 0.2079
Pendigits	0.3503 ± 0.1856	0.3471 ± 0.1851	0.3653 ± 0.2008
Page-blocks	0.0406 ± 0.0137	0.0411 ± 0.0138	0.0418 ± 0.0125
Segment	0.4079 ± 0.2187	0.4065 ± 0.2158	0.4118 ± 0.2165
Vehicle	0.4544 ± 0.1397	0.4546 ± 0.1402	0.4458 ± 0.1371
Vowel	0.4156 ± 0.2001	0.4181 ± 0.2006	0.4151 ± 0.1974
Waveform	0.4646 ± 0.1882	0.4647 ± 0.1902	0.4648 ± 0.1885

Reported values are the averages over the test folds followed by the standard deviation.

Appendix C. Rate of prediction agreement

Rate of agreement of the predictions of binary voting and adaptive voting with respect to the predictions of weighted voting is presented in Table C1.

Table C1
Rate of agreement of the predictions of binary voting and adaptive voting with respect to the predictions of weighted voting.

Data set	MLP		*k-NN*		J48	
	BV	AV	BV	AV	BV	AV
Anal-author	0.9990	0.9990	1	1	0.9888	0.9905
Balance-scale	0.9923	0.9933	0.9584	0.9827	0.9978	1
Cars	0.9084	0.9655	0.9897	0.9897	0.9768	0.9788
CMC	0.9339	0.9127	0.9829	0.9815	0.9484	0.9746
Eucalyptus	0.9476	0.9522	0.9367	0.9386	0.9481	0.9674
Glass	0.9430	0.9776	0.9776	0.9776	0.9159	0.9430
MFEAT-Fourier	0.9725	0.9739	0.9862	0.9862	0.9550	0.9612
MFEAT-Karhunen	0.9863	0.9869	0.9968	0.9968	0.9566	0.9603
MFEAT-Morpho	0.8535	0.8632	0.9889	0.9889	0.9715	0.9752
MFEAT-Zernike	0.9466	0.9536	0.9936	0.9936	0.8568	0.8990
Optdigits	0.9925	0.9943	0.9995	0.9995	0.9788	0.9780
Pendigits	0.9932	0.9946	0.9999	0.9999	0.9911	0.9910
Page-blocks	0.9769	0.9772	0.9989	0.9989	0.9986	0.9987
Segment	0.9911	0.9919	0.9965	0.9965	0.9934	0.9932
Vehicle	0.9173	0.9548	0.9898	0.9898	0.9357	0.9541
Vowel	0.8681	0.8606	0.6667	0.6667	0.9364	0.9422
Waveform	0.9996	0.9996	0.9995	0.9995	0.9924	0.9945

Test statistics are shown dependent on the used base classifier.

References

[1] J. Fürnkranz, Round robin classification, Journal of Machine Learning Research 2 (March) (2002) 721–747.
[2] J. Fürnkranz, Round robin rule learning, in: C. Brodley, A. Danyluk (Eds.), Proceedings of the 18th International Conference on Machine Learning (ICML 2001), Morgan Kaufmann, Williamstown, MA, USA, 2001, pp. 146–153.
[3] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, IEEE Transactions on Neural Networks 13 (2) (2002) 415–425.
[4] T. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, Journal of Artificial Intelligence Research 2 (July–May) (1995) 263–286.
[5] J. Friedman, Another approach to polychotomous classification, Technical Report, Department of Statistics, Stanford University, 1996.
[6] M. Moreira, E. Mayoraz, Improved pairwise coupling classification with correcting classifiers, in: C. Nedellec, C. Rouveirol (Eds.), Proceedings of the 10th European Conference on Machine Learning (ECML 1998), Springer, Chemnitz, Germany, 1998, pp. 160–171.
[7] J. Fürnkranz, Round robin ensembles, Intelligent Data Analysis 7 (5) (2003) 385–404.
[8] T.-F. Wu, C.-J. Lin, R. Weng, Probability estimates for multi-class classification by pairwise coupling, Journal of Machine Learning Research 5 (August) (2004) 975–1005.
[9] B. Quost, T. Denoeux, M.-H. Masson, Pairwise classifier combination using belief functions, Pattern Recognition Letters 28 (5) (2007) 644–653.
[10] L. Xu, A. Krzyzak, C. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Transactions on Systems, Man and Cybernetics 22 (3) (1992) 418–435.
[11] E. Hüllermeier, J. Fürnkranz, W. Cheng, K. Brinker, Label ranking by learning pairwise preferences, Artificial Intelligence 172 (16–17) (2008) 1897–1916.
[12] E. Hüllermeier, J. Fürnkranz, On predictive accuracy and risk minimization in pairwise label ranking, Journal of Computer and System Sciences, 2008, accepted for publication.
[13] S. Har-Peled, D. Roth, D. Zimak, Constraint classification for multiclass classification and ranking, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Advances in Neural Information Processing Systems 15 (NIPS 2002), MIT Press, Vancouver, British Columbia, Canada, 2002, pp. 785–792.
[14] O. Dekel, C. Manning, Y. Singer, Log-linear models for label ranking, in: S. Thrun, L. Saul, B. Schölkopf (Eds.), Advances in Neural Information Processing Systems 16 (NIPS 2003), MIT Press, Vancouver and Whistler, BC, Canada, 2003.
[15] K. Crammer, Y. Singer, Ultraconservative online algorithms for multiclass problems, Journal of Machine Learning Research 2 (January) (2003) 951–991.
[16] N. Ailon, M. Mohri, An efficient reduction of ranking to classification, Machine Learning 29 (2) (2008) 103–130.
[17] J. Fürnkranz, E. Hüllermeier, Pairwise preference learning and ranking, in: N. Lavrac, D. Gamberger, L. Todorovski, H. Blockeel (Eds.), Proceedings of the 13th European Conference on Machine Learning (ECML 2003), Springer, Cavtat, Dubrovnik, Croatia, 2003, pp. 145–156.
[18] J. Fodor, M. Roubens, Fuzzy Preference Modelling and Multicriteria Decision Support, Kluwer Academic Publishers, Dordrecht, 1994.
[19] E. Hüllermeier, K. Brinker, Learning valued preference structures for solving classification problems, Fuzzy Sets and Systems 159 (18) (2008) 2337–2352.
[20] T. Hastie, R. Tibshirani, Classification by pairwise coupling, The Annals of Statistics 26 (2) (1998) 451–471.
[21] F. Cutzu, Polychotomous classification with pairwise classifiers: a new voting principle, in: T. Windeatt, F. Roli (Eds.), Proceedings of the Fourth International Workshop on Multiple Classifier Systems (MCS 2003), Springer, Guilford, UK, 2003, pp. 115–124.
[22] B. Zadrozny, C. Elkan, Transforming classifier scores into accurate multiclass probability estimates, in: D. Hand, D. Keim, R. Ng (Eds.), Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2002), ACM, Edmonton, Alberta, Canada, 2002, pp. 694–699.
[23] A. Niculescu-Mizil, R. Caruana, Predicting good probabilities with supervised learning, in: L. De Raedt, S. Wrobel (Eds.), Proceedings of the 22nd International Conference on Machine Learning (ICML 2005), ACM, Bonn, Germany, 2005, pp. 625–632.
[24] J. Zhang, Y. Yang, Probabilistic score estimation with piecewise logistic regression, in: C. Brodley (Ed.), Proceedings of the 21st International Conference on Machine Learning (ICML 2004), ACM, Banff, Alberta, Canada, 2004, pp. 115–123.
[25] J. Marden, Analyzing and Modeling Rank Data, Chapman & Hall, London, UK, 1995.
[26] E. Hüllermeier, J. Fürnkranz, Comparison of ranking procedures in pairwise preference learning, in: B. Bouchon-Meunier, G. Coletti, R. Yager (Eds.), Proceedings of the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2004), Springer, Perugua, Italy, 2004, pp. 535–542.
[27] P. Domingos, M. Pazzani, On the optimality of the simple Bayesian classifier under zero–one loss, Machine Learning 29 (2) (1997) 103–130.
[28] A. Asuncion, D. Newman, UCI machine learning repository ⟨http://www.ics.uci.edu/~mlearn/MLRepository.html⟩, 2007.
[29] P. Vlachos, StatLib project repository ⟨http://lib.stat.cmu.edu/⟩, 2008.
[30] I. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, second ed., Morgan Kaufmann, San Francisco, CA, USA, 2005.
[31] J. Demšar, Statistical comparisons of classifiers over multiple data sets, The Journal of Machine Learning Research 7 (January) (2006) 1–30.
[32] E. Hüllermeier, J. Fürnkranz, On minimizing the position error in label ranking, in: J. Kok, J. Koronacki, R.L. de Mántaras, S. Matwin, D. Mladenic, A. Skowron (Eds.), Proceedings of the 18th European Conference on Machine Learning (ECML 2007), Springer, Warsaw, Poland, 2007, pp. 583–590.
[33] M. Evans, N. Hastings, B. Peacock, Statistical Distributions, third ed., Wiley, New York, USA, 2000.

About the Author—EYKE HÜLLERMEIER is with the Department of Mathematics and Computer Science at Marburg University (Germany), where he holds an appointment as a full professor and heads the knowledge engineering and bioinformatics Lab. He holds M.Sc. degrees in mathematics and business computing, a Ph.D. in computer science, and a Habilitation degree, all from the University of Paderborn (Germany). His research interests are focused on machine learning and data mining, fuzzy set theory, uncertainty and approximate reasoning, and applications in bioinformatics. He has published numerous research papers on these topics in respective journals and major international conferences. He is a member of the IEEE, the IEEE Computational Intelligence Society, and a board member of the European Society for Fuzzy Logic and Technology (EUSFLAT). He is on the editorial board of the journals Fuzzy Sets and Systems (area editor), Soft Computing, Advances in Fuzzy Systems, International Journal of Data Mining, Modeling and Management (associate editor), and The Open Applied Informatics Journal. Moreover, he is a co-ordinator of the EUSFLAT working group on Learning and Data Mining, the head of the IEEE CIS Task Force on Machine Learning.

About the Author—STIJN VANDERLOOY is a Ph.D. student at the Maastricht ICT Competence Centre of Maastricht University (The Netherlands). He holds a M.Sc. degree in computer science. His research interest covers the area of machine learning and its applications to biology and law enforcement. He has published in journals and international conferences on topics such as ROC analysis, performance evaluation using the AUC, reliable instance classifications, and reducing multi-class to binary classification problems.