Noname manuscript No.
(will be inserted by the editor)

On the Effect of Calibration in Classifier Combination

Antonio Bella · Cèsar Ferri · José Hernández-Orallo · María José Ramírez-Quintana

Received: date / Accepted: date

Abstract A general approach to classifier combination considers each model as a probabilistic classifier which outputs a class membership posterior probability. In this general scenario, it is not only the quality and diversity of the models which are relevant, but the level of calibration of their estimated probabilities as well. In this paper, we study the role of calibration before and after classifier combination, focusing on evaluation measures such as MSE and AUC, which better account for good probability estimation than other evaluation measures. We present a series of findings that allow us to recommend several layouts for the use of calibration in classifier combination. We also empirically analyse a new non-monotonic calibration method that obtains better results for classifier combination than other monotonic calibration methods.

Keywords Classifier combination · Classifier calibration · Classifier diversity · Probability estimation · Calibration measures · Separability measures

1 Introduction

The problem of combining multiple decisions from a set of classifiers is known as classifier combination or classifier fusion [31]. The need for classifier combination in many real applications is well-known [18][43]. On the one hand, more and more applications require the integration of models and experts that come from different sources (human expert models, data mining or machine learning models, etc.). On the other hand, it has been shown that an *appropriate* combination of several models can give better results than any of the single models alone [11][31][34][35][42], especially if the base classifiers are diverse [33].

The term *ensembles* [16][31] is used especially when the set of base classifiers are created on purpose for the problem and are homogeneous. Some ensemble techniques show an excellent performance, such as boosting [23][41][44], bagging [8][44], and randomisation [17]. If the set of classifiers is created on purpose but not homogeneous, the term ensemble is not so frequently used. Some of these techniques are stacking [46], cascading [24][29] and delegating [19]. However, in many situations, there is no on-purpose generation of a set of classifiers and we have to combine opinions from many sources (either human or machines) into a single decision. In these cases, we have no control over the set of classifiers that we have to combine and heterogeneity is very high. In this paper, we cover these three types (or degrees) of classifier combination. Since the latter case is more challenging and general, no assumption will be made about the way in which the set of classifiers have been obtained. Consequently, we will study classifier combination from an uncontrolled and heterogeneous set of models, where some (or all) of them may be machine learning models, or models that have been constructed by human experts. The only (mild) assumption is that we expect all these classifiers to be able to output a posterior probability, a reliability value or score for their prediction.

Hence, given a set of (heterogeneous) classifiers, it is very important to assess and use their individual quality for a proper classifier combination. Typically, an overall weighting is used in such a way that more

A. Bella · C. Ferri · J. Hernández-Orallo · M.J. Ramírez-Quintana
DSIC-ELP, Universitat Politècnica de València, Camí de Vera s/n, 46022 Valencia, Spain
Tel: +34 96 387 7007 Ext: {83502, 83505, 73585, 73586}
Fax: +34 96 387 73 59
E-mail: {abella, cferri, jorallo, mramirez}@dsic.upv.es

reliable classifiers are given more weight than other less reliable classifiers. The way in which 'reliability' is measured and the way in which it is used in the combination make up different weighted combination schemes (see, e.g., [31]). In the same way, diversity has typically been analysed in terms of qualitative measures, over a given dataset, such as disagreement, Q-statistic, Kappa statistic, etc. [33][32].

When classifiers are considered as rankers or probability estimators there are many more available options. If we understand probabilities as an indicator of reliability, we can have a *double-weighting* combination scheme, where overall classifier accuracy is used as a *first weight* while estimated class probabilities are used as a *second weight*. For instance, given two classifiers a and b where the accuracy (or any other performance measure) of a is worse than the accuracy of b, we can still have that, for a particular item i, classifier a may be very sure (with extreme probabilities) while classifier b may be more uncertain (with medium probabilities). Of course, for other items, this might be the other way round. If probabilities are well estimated, this *second weight* will give different levels of credit to each classifier depending on the example at hand, and might be more precise than the overall weight given to each classifier.

Therefore, the key point in this combination scheme is the quality of probability estimations. When the set of classifiers is heterogeneous and originates from different sources, we cannot assume that these estimations are evenly accurate and reliable. In fact some classifiers may output more extreme probabilities (closer to 0 or to 1) than others, meaning that they will have more (second) weight in the combination. This is the typical issue of integrating opinions from several experts, when some experts express less or more confidence than they really have. We say these experts are *uncalibrated*. It is well known [30] that bad classifiers in an ensemble will probably deteriorate the overall result if we combine label (class) predictions. But it is not so well-known that when we combine estimated probabilities, a single very bad classifier with very extreme probabilities may have a devastating effect.

It is then of the utmost importance to analyse the effect of calibration in classifier combination by studying several combination layouts and several calibration methods, and their impact on the quality of the combined classifier, according to several evaluation metrics. This is the goal of the paper.

The paper is structured as follows. In Section 2, we give further motivation for this study and we also point out to related (but partial) analyses on this issue in the literature. Section 3 summarises the most common evaluation measures and calibration methods. Section 4 includes a conceptual study on calibration and classifier combination, using several examples and identifying several important factors. Section 5 presents the experimental evaluation of the previous analysis. In Section 6, we show that monotonic calibration techniques are non-monotonic when applied to more than two classes. This justifies the application and analysis of a new non-monotonic calibration technique known as *Multivariate Similarity-Binning Averaging*. The whole picture is studied in Section 7 with several combination layouts. Finally, Section 8 gives an overall view of the messages that can be conveyed from this paper, leading to the conclusions and future work.

2 Context and Objectives

Classifier combination has been extensively analysed in the past few decades, establishing very important results about the number of classifiers, their diversity, the combination method, etc. In this paper, we focus on one factor that has not been properly addressed to date: the role of probability calibration in classifier combination.

This role has many aspects: different probability calibration measures, different calibration methods, different layouts where calibration has to be arranged, etc. In addition, we must consider the relation with other fundamental issues in classifier combination, such as classification quality and diversity (which can be evaluated by different families of evaluation measures). However, only a few research efforts have been done for addressing some of these issues.

There are, for instance, some approaches which use combination and calibration, but generally with a very specific layout. For example, in [38], the *Expectation-Maximization algorithm* was used to modify the weights of a Bayesian Model Averaging [28] method and to obtain a calibrated ensemble, but the effect of calibration methods before the combination was not studied.

Caruana et al [12] also used calibration and combination together. The experimental setting in [12] analysed many other factors altogether but only included one calibration method (Platt), it was restricted to binary datasets, and the double weighting effect was not evaluated (a uniform weighting was used for Bayesian Averaging).

In Bennett's Ph.D. thesis [5], a methodology is introduced to build a metaclassifier for classifying text documents by combining estimated probabilities of base classifiers and using reliability indicators. These reliability indicators are variables with additional information, not mere probabilities, and are application specific.

Brümmer's Ph.D. thesis [10] focuses on speaker and language recognisers. Instead of calibrating the probabilities estimated by a classifier, he calibrates the *log-likelihood-ratio* scores. However, the calibration and combination methods studied are always affine (i.e., linear).

The combination of classifiers and their calibration has also been indirectly addressed when trying to adapt binary calibration methods to multiclass calibration methods, since multiclass calibration can be tackled by a combination of binary calibration methods. For instance, in Gebel's Ph.D. thesis [26], there is a study of several univariate calibration methods and their extensions to multiclass problems, but only individually, i.e., without combining them. Moreover, Gebel introduced *Dirichlet calibration* as a multivariate calibrator that is applicable to multiclass problems directly, but its poor overall results make it only recommendable for datasets that have a balanced or slightly imbalanced class distribution.

In the end, this paper analyses the overall issue, bringing calibration to the stage of classifier combination as another key dimension of study, jointly with the well-known properties of classification accuracy and diversity. A general analysis of classifier fusion (or combination) and calibration cannot be found in the literature.

Therefore, the objective of this paper is to undertake this analysis. Along the way, this study introduces different contributions that can be summarised as follows:

- A conceptual analysis on how calibration affects the combination of classifiers. This analysis is performed in terms of how classifier probability distributions relate to the combination results depending on the separation of the class distribution, the calibration and diversity of the base classifiers.
- An extensive experimental comparison of the effect of calibration on the combination of classifiers, using many different layouts (calibration before, after, and before and after combination), many different weighting schemes, several calibration methods, and several performance metrics.
- The analysis of a new calibration technique: Multivariate Similarity-Binning Averaging (SBA), recently introduced in [4], which is designed to be non-monotonic while still preserving independence in such a way that its results for classifier combination excel over those of other calibration methods. This will be shown in a complete battery of experiments.
- A summary of findings and results from the previous items, and some recommendations about how to use calibration in classifier combination.

The overall contribution of this paper is to provide a better understanding of the role and possibilities of calibration for classifier combination, as well as the way all this should be arranged in order to obtain appropriate results.

3 Classifier Calibration and Evaluation

Given a dataset T, n denotes the number of examples, and c the number of classes. The target function $f(i,j) \to \{0,1\}$ represents whether example i actually belongs to class j. Also, $n_j = \sum_{i=1}^{n} f(i,j)$ denotes the number of examples of class j and $p(j) = n_j/n$ denotes the prior probability of class j. A crisp classifier outputs the predicted class j for each example while a soft classifier produces a probability (or score) for each pair of example and class. Given a soft classifier l, $p_l(i,j)$ represents the estimated probability that example i belongs to class j taking values in [0,1], whereas a score $s_l(i,j)$ is an indicator of the reliability about example i being of class j. Unlike probabilities, scores are not bounded by the $[0,1]$ interval. In what follows we will assume that we deal with soft classifiers outputting probabilities. A soft classifier can be turned into a crisp classifier issuing decisions by the use of thresholds. For instance, if we set the threshold θ_j for class j then we have that class j is predicted for example i when $p(i,j) \geq \theta_j$. For two classes, since probabilities are complementary, we only need one threshold.

Calibration is defined as the degree of approximation of the predicted probabilities to the actual probabilities. If we predict that we are 99% sure, we should expect to be right 99% of the time. More precisely, a classifier is perfectly calibrated if, for a sample or bin of examples with predicted probability p for the positive class, the expected proportion of positives is equal to p. Formally, for any $B_r \subseteq T$ such that $p_l(i,j) = r$ for all $i \in B_r$ then $\frac{\sum_{i \in B_r} f(i,j)}{|B_r|} = r$. Note that this definition only says when a classifier is perfectly calibrated but does not give a range of values between perfect and worst calibration. We will see in Section 3.1 that calibration measures usually relax the condition for bin formation in order to give a gradual measure.

Given a calibration method which modifies the probabilities (or converts scores into probabilities), we denote the (supposedly better calibrated) probability that example i belongs to class j by $p_l^*(i,j)$. Note that accuracy and calibration, although dependent, are very different things. For instance, a binary classifier that

always assigns a 0.5 probability to its predictions is perfectly calibrated for a balanced dataset, but its expected accuracy is a poor 0.5. However, a very good binary classifier can be uncalibrated if correct positive (respectively negative) predictions are accompanied by relatively low (respectively high) probabilities, e.g., a classifier which is almost always correct but its probabilities range between 0.45 and 0.55.

A set of L classifiers will be denoted by $l_1, l_2, \ldots, l_L$. In order to simplify the notation, sometimes we will use the indice $k \in 1..L$ to denote the classifier l_k. Finally, $\tilde{p}(i,j)$ (respectively, $\tilde{s}(i,j)$) denotes the estimated probability (respectively, the score) that example i belongs to class j given by a combination method over the L classifiers.

When the number of classes is 2 we use the special symbols $\oplus$ and $\ominus$ to represent the positive class ($j = 1$) and the negative one ($j = 2$), respectively. Also, in the binary case, we will only refer to the positive class, and we will denote the target function, the score, the estimated probability, and the calibrated probability of an example i as $f_l(i,\oplus)$, $s_l(i,\oplus)$, $p_l(i,\oplus)$ and $p_l^*(i,\oplus)$ or simply $f_l(i)$, $s_l(i)$, $p_l(i)$ and $p_l^*(i)$. For the sake of readability, we will omit the subindex l when we refer to a single classifier.

3.1 Evaluation Measures

Classifiers can be evaluated according to several performance metrics. These can be classified into three groups [20]: measures that account for a qualitative notion of error (such as accuracy or the mean F-measure/F-score), metrics based on how well the model ranks the examples (such as the Area Under the ROC Curve (AUC)) and, finally, measures based on a probabilistic understanding of error (such as mean absolute error, mean squared error (Brier score), LogLoss and some calibration measures).

Accuracy is the best-known evaluation metric for classification and is defined as the percentage of correct predictions. However, accuracy is very sensitive to class imbalance. In addition, when the classifier is soft, accuracy depends on the choice of a threshold. Hence, a good classifier with good probability estimations can have low accuracy results if the threshold that separates the classes is not chosen properly.

Of the family of measures that evaluate ranking quality, the most representative one is the *Area Under the ROC Curve (AUC)*. For two classes, this is interpreted as the probability that given one positive example and one negative example at random, the classifier ranks the positive example above the negative one (the Mann-Whitney-Wilcoxon statistic [22]). AUC is clearly a measure of separability since the lower the number of misranked pairs, the better separated the classes are. Although ROC analysis is difficult to extend to more than two classes ([21]), the AUC has been extended to multiclass problems effectively by approximations. In this paper, we will use Hand & Till's extension [27], which is based on an aggregation of each class against each other, by using a uniform class distribution.

Of the last family of measures, *Mean Squared Error (MSE)* or, for two classes, *Brier Score* [9] penalises strong deviations from the true probability:

$$MSE = \frac{\sum_{j=1}^{c}\sum_{i=1}^{n}(f(i,j) - p(i,j))^2}{n \cdot c} \quad (1)$$

Although MSE was not a calibration measure originally, it was decomposed by Murphy [36] in terms of calibration loss and refinement loss. For that, the dataset T is segmented into m bins (i.e., subsets of equal size), with B_t being the elements of bin t. Bins must be constructed as a sequential partition of the examples ordered by estimated probability (or score):

$$MSE = \frac{\sum_{j=1}^{c}\sum_{t=1}^{m}\sum_{i\in B_t}|B_t| \cdot (p(i,j) - \bar{f}_t(i,j))^2}{n \cdot c} \quad (2)$$
$$- \frac{\sum_{j=1}^{c}\sum_{t=1}^{m}|B_t| \cdot (\bar{f}_t(j) - \bar{f}(j)) + \bar{f}(j) \cdot (1 - \bar{f}(j))}{n \cdot c}$$

where $\bar{f}_t(j) = \sum_{i\in B_t}\frac{f(i,j)}{|B_t|}$ and $\bar{f}(j) = \sum_{i=1}^{n}\frac{f(i,j)}{n}$. The first term measures the calibration (denoted by *MSEcal*) of the classifier while the rest of the expression measures other components that are grouped under the term *refinement* (denoted by *MSEref*). The refinement component indicates the usefulness of each prediction for distinguishing (i.e., separating) the classes [14]. Note that refinement only depends on the order of the examples, like AUC. The problem of measuring calibration in that way is that the test set must be split into several segments or bins. If too few bins are defined, the actual probabilities $\bar{f}_t(j)$ are not properly detailed to give an accurate evaluation. In fact, when we only consider one bin we talk about *global* calibration, instead of the common (local) calibration. If too many bins are defined, the actual probabilities are not properly estimated. A partial solution is to make the bins overlap.

A calibration measure based on overlapping binning is *CalBin* [13]. For each class, all cases i must be ordered by their estimated probability $p(i,j)$, as in the MSE decomposition above. The 100 first elements (i from 1 to 100) are taken as the first bin. Next, the percentage of cases of class j in a bin t is $\bar{f}_t(j)$ defined above.

The error for this bin is $\sum_{i\in 1..100} |p(i,j) - \bar{f}_t(j)|$. The second bin with elements (2 to 101) is used to compute the error in the same way. At the end, the errors are averaged. Formally:

$$CalBin(j) = \frac{1}{n-s}\sum_{b=1}^{n-s}\sum_{i=b}^{b+s-1}\left| p(i,j) - \sum_{i=b}^{b+s-1}\frac{f(i,j)}{s}\right| \quad (3)$$

Instead of 100 for the size of the bin (as [13] suggests) we set a different bin length, $s = n/10$, to make it more size-independent.

3.2 Calibration Methods

In this paper, we will empirically analyse the most commonly used calibration methods: binning averaging, Platt's method and PAV calibration. There are other methods based on assignment values [25], Bayesian approaches using asymmetric distributions [6][5], and other more elaborate approaches, such as Dirichlet calibration [26], but their performance, in general, is worse than that of the three methods above. For more details, we refer the reader to [3] where a survey of calibration methods can be found.

Binning averaging was proposed by [47] as a method for binary classifiers where a (validation) dataset is split into bins in order to calculate a probability for each bin. Specifically, this method is based on sorting the examples in decreasing order by their estimated probabilities (or scores) and dividing the set into m bins. Thus, each test example i is placed into a bin t, $1 \leq t \leq m$, according to its probability estimation. Then the corrected probability estimate for i ($p^*(i)$) is obtained as the proportion of instances in t of the positive class, i.e., $\bar{f}_t(\oplus)$.

Platt [37] presented a parametric approach for fitting a sigmoid function that maps SVM predictions to calibrated probabilities. The idea is to determine the parameters A and B of the sigmoid function $p^*(i)=\frac{1}{1+e^{A\cdot p(i)+B}}$ that minimises the negative log-likelihood of the data, that is: $argmin_{A,B}\{-\sum_i f(i)log(p^*(i)) + (1-f(i))log(1-p^*(i))\}$. This two-parameter minimisation problem can be performed by using an optimisation algorithm, such as *gradient descent*. Platt proposed the use of either cross-validation or a hold-out set for deriving an unbiased sigmoid training set for estimating A and B.

In the *Isotonic Regression* [40] method, the calibrated predictions are obtained by applying a mapping transformation that is isotonic (monotonically increasing), known as the pair-adjacent violators (PAV) algorithm [2]. The first step in this algorithm is to order the n elements decreasingly according to estimated probability and to initialise $p^*(i) = f(i)$. The idea is that calibrated probability estimates must be a decreasing sequence, i.e., $p^*(1) \geq p^*(2) \geq \ldots \geq p^*(n)$. If this is not the case, for each pair of consecutive probabilities, $p^*(i)$ and $p^*(i+1)$, such that $p^*(i) < p^*(i+1)$, the PAV algorithm replaces both of them by their probability average, that is, $a \leftarrow \frac{p^*(i)+p^*(i+1)}{2}$, $p^*(i) \leftarrow a$, $p^*(i+1) \leftarrow a$. This process is repeated (using the new values) until an isotonic set is reached.

Not all calibration methods are equally suitable for every type of problem. For instance, if we do not have enough validation data (or we need to calibrate with the training data), the PAV algorithm is prone to overfitting. Also, some of these methods may be more appropriated for highly imbalanced data or may work better for multiclass problems (using a one-vs-all or one-vs-one approach). Nonetheless, to our knowledge, there is no comprehensive study of which method can be better depending on the kind of problem.

3.3 Monotonicity and Multiclass Extensions

The three calibration methods described above are monotonic; they do not change the rank (order) of the examples according to each class estimated probability. In fact, Platt's method is the only one that is strictly monotonic, i.e., if $p(i_1) > p(i_2)$, then $p^*(i_1) > p^*(i_2)$, implying that AUC and refinement are not affected. In the other two methods, ties may be created (i.e., $p^*(i_1) = p^*(i_2)$ for some examples i_1 and i_2 where $p(i_1) > p(i_2)$). Refinement reaches a maximum (worst possible value) when all examples tie, and no separability takes place. In general, refinement increases (worsens) for the binning averaging and the PAV methods if ties are created.

Monotonicity will play a crucial role in understanding what calibration does before classifier combination. However, as we will see in Section 6, the multiclass extension of calibration methods does not preserve monotonicity. In addition, apart from overfitting, there is no reason to impose monotonicity to a calibration method, which, in the most general case, is a transformation over the scores or probabilities that leads to good probability estimation. This will motivate the analysis of a recently introduced non-monotonic calibration method called SBA.

Based on the concept of monotonicity, we propose a taxonomy of calibration methods (Figure 1) including classical calibration methods (PAV, Binning and Platt), the SBA method and Brümmer's *affine fusion and calibration* methods [10]. We are interested in calibration methods that lead to better local calibration because global calibration is useless to get better individual re-

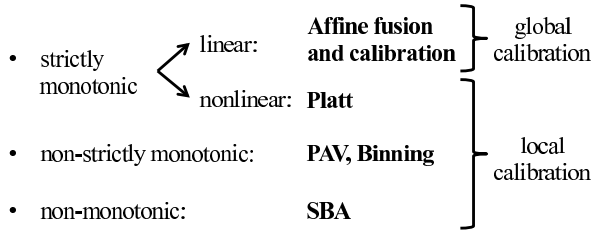

Fig. 1 Taxonomy of calibration methods in terms of monotonicity (strictly monotonic, non-strictly monotonic, or non-monotonic methods) and linearity (linear or nonlinear methods). We have also indicated the methods that can be used for global and local calibration.

liability assessment for combination. Consequently, we will not use Brümmer's method in this paper.

Another important issue is whether the calibration methods are binary or multiclass. The three methods presented in Section 3.2 were specifically designed for two-class problems. For the multiclass case, Zadrozny and Elkan [47] proposed an approach that consists in reducing the multiclass problem into a number of binary problems. A classifier is learnt for each binary problem and, then, its predictions are calibrated. Some studies have compared the *one-against-all* and the *all-against-all* schemes, concluding in [39] that the *one-against-all* scheme performs as well as the *all-against-all* schemes. Therefore, in this paper, we will use the *one-against-all* approach for our experimental analysis because its implementation is simpler.

4 The Relation between Calibration and Combination

In this section, we analyse the relation between model calibration and the performance of the classifier combination. For simplicity, we restrict our conceptual analysis to binary cases. Similar relations are expected to be found if we consider the probability distribution of each class against another (so having $c \times (c-1)$ distributions). In any case, the experimental analysis in Section 5 will be performed on multiclass datasets as well.

4.1 Weighted Average Combination

One of the most common methods of classifier combination is *Bayesian Model Averaging* [28]. It consists in weighting each classifier, giving more credit to more reliable sources. However, this rationale does not necessarily entail the best combination [30][31]. An alternative (and generalised) option is the weighted average combination [31], using probabilities:

Definition 1 Weighted Average Combination. The estimated probability of an item i belonging to class j given by a weighted average combination of L classifiers is

$$\tilde{p}(i,j) = \sum_{k=1}^{L} w_k p_k(i,j) \tag{4}$$

We assume $\sum_{k=1}^{L} w_k = 1$. Formula (4) defines a fusion scheme that is a linear combination of the classifier outputs and can be instantiated to more specific schemas depending on how w_k and p_k are chosen. In general, the use of a performance (or overall reliability) weight per classifier w_k is justified because some classifiers are more reliable than others. However, a proper calibration would give each prediction its proper weight depending on the reliability of $p_k(i,j)$ (high reliability for $p_k(i,j)$ closer to 0 and 1, and lower reliability for $p_k(i,j)$ closer to 0.5 or to the class proportion for imbalanced problems). This use of w_k and p_k at the same time is what we refer to as *double weighting*.

Example 1 Two probabilistic classifiers l_1 and l_2 are evaluated over a dataset with 8 examples as shown in Table 1, and combined using weights $w_1 = 0.75$ and $w_2 = 0.25$. The top three rows show their individual predictions and their combination. The mid three rows show the results with two new classifiers l_1^* and l_2^*, which have been obtained from l_1 and l_2 by using a strictly monotonic calibration method over another dataset (their calibration is better but not perfect). The bottom three rows show the results with two new classifiers l_1^{pav} and l_2^{pav}, which have been obtained from l_1 and l_2 by using PAV (a non-strictly monotonic calibration method) over the same dataset (so their calibration is perfect). All the accuracies are calculated with a threshold of 0.5, and when the probability is exactly 0.5 the example is considered half a correct classification.

Example 1 shows different results depending on the degree of calibration of these two classifiers. In this example, we see that weights can counter-effect probabilities (and vice versa), as we see in example e_6 (which is correctly classified by $\tilde{p}$) and examples e_2 and e_8 (which are wrongly classified by $\tilde{p}$). So, using both weights and good probabilities entails a "double-weighting", which in some cases might be beneficial but in other cases might not. Looking at the extreme cases, with very bad probabilities, the weight w_k should be used alone (as in weighted majority voting) and, with perfect probabilities, the weights should not be used.

In order to better understand the relation between weights and probabilities, we firstly need to understand the meaning of the weights. There are many ways of

Table 1 Results of two classifiers and their combination corresponding to Example 1. Top three rows: without calibration. Three rows in the middle: by applying a strictly monotonic calibration method. Bottom three rows: by applying a non-strictly monotonic calibration method. Threshold is set on 0.5 to calculate the accuracy (last column).

	Examples								Acc.
	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	
True class	-	+	+	+	-	-	+	-	
p_1	0.4	0.6	0.8	0.4	0.6	0.52	0.6	0.45	5/8
p_2	0.9	0.1	0.2	0.2	0.2	0	0.2	1	2/8
$\tilde{p}$	0.53	0.48	0.65	0.35	0.5	0.39	0.54	0.59	3/8
p_1^*	0.2	0.8	0.9	0.2	0.8	0.6	0.8	0.3	5/8
p_2^*	0.6	0.4	0.45	0.45	0.45	0.1	0.45	0.9	2/8
$\tilde{p}^*$	0.3	0.7	0.79	0.26	0.71	0.48	0.71	0.45	6/8
p_1^{pav}	0.25	0.67	1	0.25	0.67	0.5	0.67	0.25	5.5/8
p_2^{pav}	0.57	0.5	0.57	0.57	0.57	0	0.57	0.57	4.5/8
$\tilde{p}^{pav}$	0.33	0.63	0.89	0.33	0.64	0.38	0.64	0.33	5/8

calculating weights. A very common option is to estimate the accuracy on a validation dataset D, followed by a normalisation [31], i.e., if acc_k is the accuracy of model l_k on D, then $w_k = \frac{acc_k}{\sum_{m=1}^{L} acc_m}$. If we use AUC (or MSE) as a measure, the question of whether a double weighting is going to be too drastic depends on how the weights are derived from these measures. Note that with a normalisation we ensure that the weights add up to one, but the weights may still be too similar or too different. For instance, a weight equal to the AUC is an option, but since AUC=0.5 means random behaviour, almost all classifiers will generally have values between 0.5 and 1 that, after normalisation, will be fairly similar. Hence, probably the GINI index (which equals $(AUC-0.5)\times 2$) would be a better option. In the same way, using the MSE, the formula $(1-MSE)$ is a natural option, but a more extreme $1/MSE$ could also be considered. Table 2 shows the definition for the five weights with which we are going to instantiate Equation 4.

Table 2 Different methods to calculate weights.

Method	Definition
WCUnif	$w_k = \frac{1}{L}$
WCAcc	$w_k = \frac{acc_k}{\sum_{m=1}^{L} acc_m}$
WCAUC	$w_k = \frac{AUC_k}{\sum_{m=1}^{L} AUC_m}$
WCMSE	$w_k = \frac{(1-MSE_k)}{\sum_{m=1}^{L} (1-MSE_m)}$
WCGINI	$w_k = \frac{max(0,(AUC_k-0.5)\times 2)}{\sum_{m=1}^{L} max(0,(AUC_m-0.5)\times 2)}$
WCIMSE	$w_k = \frac{(1/MSE_k)}{\sum_{m=1}^{L} (1/MSE_m)}$

Another problem of weights is that they may overfit. Consequently, in some experimental analyses [30][31], there are cases where the use of a *uniform weighting* (WCUnif) gives better results.

There are many open questions when mixing together probabilities and weighted combinations. Are both things redundant or even incompatible? Is calibration a good idea to get better probability estimations? If calibration is used, would weights become useless?

4.2 Probability Densities and Classifier Combination

Before examining the effect of calibration on classifier combination (Section 4.3), we need to understand how estimated probabilities can be distributed and how this distribution affects the combination of classifiers.

Figure 2 shows the probability densities (for the positive class, $p(i,\oplus)$) for three different classifiers (J48, Random Forest and Naïve Bayes, built with Weka [45]) and the *credit* dataset from the UCI repository [7].

Typically, the positive cases cluster around a high probability and the negative cases cluster around a low probability. When the two clusters are more distant and better delineated (with a thinner shape, as shown in the top chart of Figure 2) there is better separability (and, hence, higher AUC). Also, in these charts, calibration can be easily calculated because each bin in the histogram should have a proportion of positive examples equal to its x-axis value when the classifier is well calibrated. For instance, perfectly calibrated classifiers should have 5% of positive examples and 95% of negative examples in the bin 0.05, 10% of positive examples and 90% of negative examples in the bin 0.10, and so on.

In the rest of this section, instead of working with empirical distributions as in Figure 2, we will use a parametric modelling of the density functions. If we were using scores instead of probabilities we could use Gaussians to model the score distribution. Since we are dealing with probabilities (which are always between 0 and 1), we will model them via a truncated normal distribution[1]. We will use the notation $n^{\oplus}$, $\mu^{\oplus}$ and $\sigma^{\oplus}$ for the number of positives, the mean of the estimated probabilities for the positives and the deviation of the estimated probabilities for the positives, respectively. Similarly, we will use the notation $n^{\ominus}$, $\mu^{\ominus}$ and $\sigma^{\ominus}$ for the negatives. Note that, although there are cases where probabilities do not strictly follow a (truncated) normal distribution, the aggregation of several non-normal distributions typically converges to a normal distribution.

[1] This distribution has been used to model probabilities for binary cases in the *probit model* or in *truncated regression* [1].

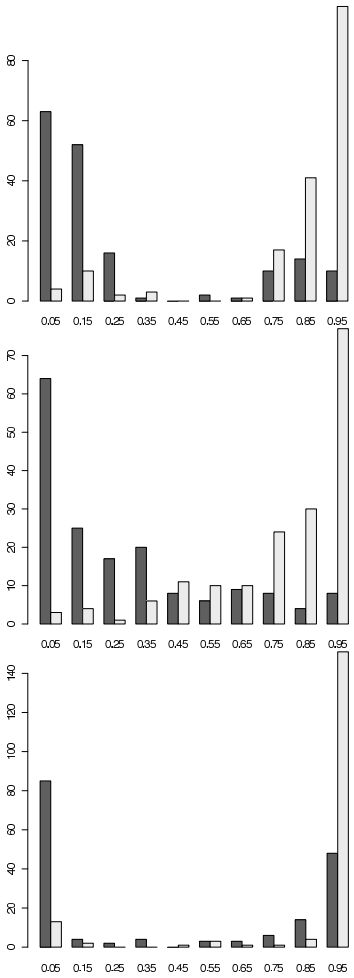

Fig. 2 Probability densities for positive (clear grey) and negative (dark grey) classes, using J48 (top row), Random Forest (centre row), and Naïve Bayes (bottom row) in Weka for the *credit* dataset from the UCI repository.

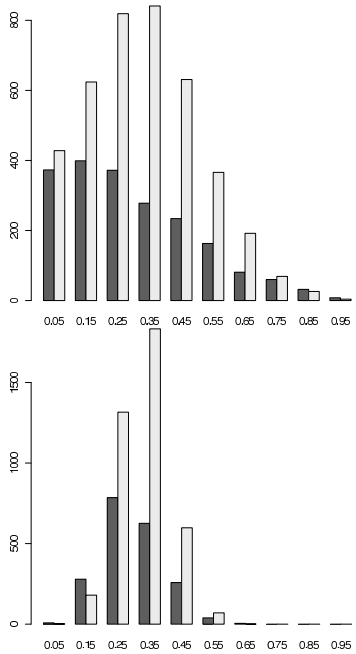

Fig. 3 Probability densities for positive (clear grey) and negative (dark grey) examples. Top: a single classifier (*Sing*) with parameters (of the normal distributions): $n^{\oplus} = 4000$, $\mu^{\oplus} = 0.3$, $\sigma^{\oplus} = 0.2$, $n^{\ominus} = 2000$, $\mu^{\ominus} = 0.15$ and $\sigma^{\ominus} = 0.3$. Bottom: the result of the combination *Com*, using uniform weights, of 5 *independent* classifiers such as *Sing*.

Therefore, at least for the combined model, this representation is not a strong working hypothesis helping us to conceptually analyse the effect of combining several classifiers. Figure 3 (top) shows a classifier modelled with two truncated normal distributions with parameters: $n^{\oplus} = 4000$, $\mu^{\oplus} = 0.3$, $\sigma^{\oplus} = 0.2$, $n^{\ominus} = 2000$, $\mu^{\ominus} = 0.15$ and $\sigma^{\ominus} = 0.3$. Figure 3 (bottom) shows the result of combining five different classifiers with the same parameters as the one in Figure 3 (top). The results for several metrics are shown for the columns *Sing* and *Com* in Table 3.

It is easy to show that combining independent classifiers that follow normal distributions leads to a combined classifier whose positive (respectively negative) mean is the (weighted) average of the positive (respectively negative) mean of the base classifiers, but the deviation is usually lower. This means that, by using a weighted average combination, the distributions are narrowed, which implies that the combination usually improves in terms of separability (provided the original classifiers were better than random, i.e., the positive mean was greater than the negative mean). This is the general picture. Can we say something more specific when we look at calibration? This is what we see next.

4.3 Calibration and Classifier Combination

A naïve view of the effect of calibration in combination would conclude that the better calibrated a classifier is, the better the reliability of its probability estimations is and, hence, the better the combination will be. However, the relationship between classifier calibration and combination is a bit more complex. When we say better, we need to be more precise about the evaluation metric that we are using.

Figures 3, 4, and 5 present an example illustrating the effect of calibration and combination over several performance measures whose results are shown in Table 3. We study the performance of combining several classifiers modelled by a normal distribution (Figure 3, top), and the role of calibration in this process. We analyse six different scenarios, a single classifier *Sing*, Platt's calibration of this classifier *SingCal*, raw combination of five classifiers *Com*, Platt's calibration after the combination of these 5 classifiers *Com+Cal*, combi-

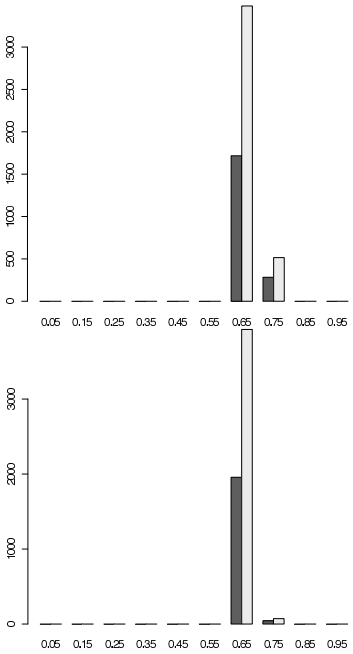

Fig. 4 Probability densities for positive (clear grey) and negative (dark grey) examples. Top: Platt's calibration (*SingCal*) applied to the single classifier *Sing*. Bottom: Platt's calibration (*Com+Cal*) applied after the combination *Com* of the original classifier.

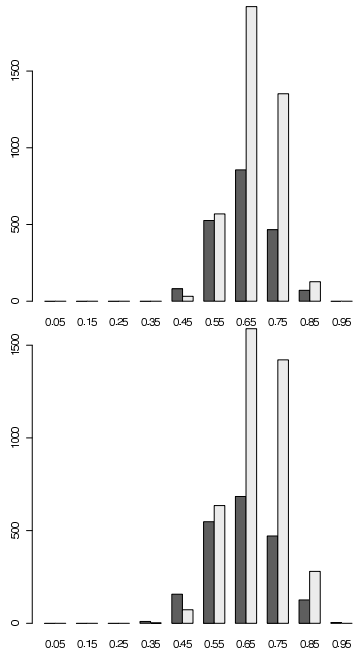

Fig. 5 Probability densities for positive (clear grey) and negative (dark grey) examples. Top: Combination (*Cal+Com*) of the 5 calibrated classifiers *SingCal*. Bottom: Platt's calibration (*Cal+Com+Cal*) applied to the combination *Cal+Com*.

Table 3 Results for several measures of the classifiers of Figures 3, 4, and 5.

	Sing	*Com*	*SingCal*	*Com +Cal*	*Cal +Com*	*Cal +Com +Cal*
MSE	0.37	0.34	0.22	0.22	0.22	0.21
MSE cal.	0.15	0.13	8e-4	1e-4	2.8e-3	3.7e-3
MSE ref.	0.22	0.21	0.22	0.22	0.21	0.21
AUC	0.56	0.60	0.56	0.61	0.60	0.61
Acc.(0.5)	0.39	0.34	0.67	0.67	0.67	0.68
CalBin	0.37	0.37	0.05	0.06	0.05	0.04

nation of five classifiers previously calibrated *Cal+Com*, and finally, Platt's calibration after combination of five classifiers previuosly calibrated *Cal+Com+Cal*.

Figure 4 shows a postprocessing using Platt's calibration over *Sing* (top plot) and *Com* (bottom plot). Finally, the plots on Figure 5 present a similar process but now we calibrate the base classifiers before their combination, without a post-calibration (top plot) and with a post-calibration (bottom plot). Since AUC is not very high in this case (separation between the classes is low), the probabilities are highly condensed in a small area of the graph. Even though Platt's calibration is not a linear scaling and we also have some truncation here, we see that there are important differences in accuracy and calibration between the two charts, but the AUC of the combination is almost the same for both cases.

A possible explanation is that monotonic transformations preserve the ranking and hence have a limited effect on the results (in terms of AUC). In addition, again we see that the result of the combination after calibrating the base classifiers does not necessarily produce a calibrated result. Evaluating the result of the combination with measures that take calibration into account, such as MSE or, indirectly, accuracy, would lead to wrong conclusions about the quality of classifiers. This reinforces the idea of AUC as the appropriate measure of combination quality, since other measures are affected by the level of calibration (and can also be improved by a post-calibration).

Example 2 below gives more clues the effect of calibration before and/or after combination, as a specific example of a much more complete and comprehensive battery of experiments that we will do in the following section. Now we focus on several classifiers with different calibration degrees.

Example 2 Consider a problem with 500 positive examples and 1000 negative examples and a diverse set of five classifiers whose parameters for the normal distributions (for the positives, $\mu^{\oplus}$ and $\sigma^{\oplus}$, and for the negatives, $\mu^{\ominus}$ and $\sigma^{\ominus}$) are shown in Table 4.

Note that we have diversity of base classifiers: not only are they independent, but they also have quite dif-

Table 4 Parameters of the normal distributions (for the positive and negative examples) of five classifiers.

#	$\mu^{\oplus}$	$(\sigma^{\oplus})^2$	$\mu^{\ominus}$	$(\sigma^{\ominus})^2$
1	0.4	0.04	0.3	0.07
2	0.5	0.3	0.3	0.2
3	0.2	0.1	0.3	0.2
4	0.8	0.04	0.2	0.3
5	0.8	0.3	0.7	0.3

ferent distributions. There are good classifiers (such as the fourth classifier) and bad classifiers (the third classifier is even worse than random). The best single classifier has an AUC of 0.87. We now consider the following calibration and combination layouts: the combination of the base classifiers (*Com*), the combination of the calibrated base classifiers (*Cal+Com*), the calibration of the combination result (*Com+Cal*) and the calibration of the combination of the calibrated base classifiers (*Cal+Com+Cal*). The results are presented in Table 5.

Table 5 Results for several calibration and combination methods in Example 2.

Method	MSE	MSEcal	MSEref	AUC	Acc	CalBin
Com	0.19	0.03	0.17	0.80	0.76	0.16
Cal+Com	0.15	0.07	0.08	0.96	0.80	0.25
Com+Cal	0.17	3e-3	0.16	0.80	0.76	0.05
Cal+Com+Cal	0.07	3e-3	0.07	0.96	0.90	0.03

This example shows the relevance of calibrating before, since only with a proper pre-calibration can we align good and bad classifiers in an optimal way. However, although calibration and combination entails an average of the means and a reduced variance (which generally implies better AUC), *this does not mean that combining perfectly calibrated classifiers generates a perfectly calibrated combination* (see the $CalBin$ result obtained by the $Cal + Com$ method).

Finally, with respect to the accuracy measure, when classifiers are well-calibrated before the combination (and classes are relatively well separated by the classifiers), the resulting means after combination will be placed on either side of the centre (the prior class proportion). If this centre value (e.g., 0.5 for a balanced datasets) is used as a threshold, then accuracy will be high. This suggests that calibration has to be considered as a necessary option before combination to increase accuracy, and not only after combination (as Table 5 shows). Nevertheless, all this is subject to choosing a good threshold, which does not need to be the prior class proportion if the combination is not well calibrated.

In summary, *the AUC measure is chosen as a reference for the quality of the combination*, since calibration measures for the combination will generally not be good (including MSE) and accuracy will greatly depend on a good threshold choice.

We now have a better understanding of how calibration affects the combination and we have identified the key factors involved: performance measures, use of weights, measure used to derive the weights, calibration monotonicity and moment of calibration (before and/or after combination). These and other issues are addressed through an experimental analysis below.

5 Experimental Analysis

This section provides a comprehensive experimental analysis about the effect of calibration and combination, focusing on the factors identified in the previous section.

5.1 Experimental Settings

For the experimental evaluation, we implemented the evaluation measures and calibration methods (the PAV algorithm, Platt's method, and binning averaging with 10 bins) presented in Sections 3.1 and 3.2. We also defined all the weighted combination schemes that use the weights shown in Table 2.

To simulate a diverse set of base classifiers that come from different sources, we used four different methods for classification implemented in the data mining suite WEKA [45] (with their default parameters) to construct several models for each problem: J48 (a C4.5 implementation), Logistic (a logistic regression implementation), IBk ($k = 10$) (a k-NN implementation) and NaïveBayes. An additional random and uncalibrated classifier (Random) was added (when necessary) to the experiments in order to compare what happens when one of the base classifiers is bad. A uniform random distribution was used to decide which probability was set to 1 or 0. This random classifier has very bad calibration, since probabilities are either 0 or 1. We selected 30 (small and medium-sized) datasets (Table 6) from the UCI repository [7]. A total of 100 repetitions were performed for each dataset and classifier. In each repetition, each dataset was split randomly into four different subsets (as can be seen in Figure 6): one for training, two for validation and one for test (25% of the instances for each set).

The training set (Train) was used to learn each classifier. One validation set (Val1) was used, in case, to calibrate the probabilities of the base classification mod-

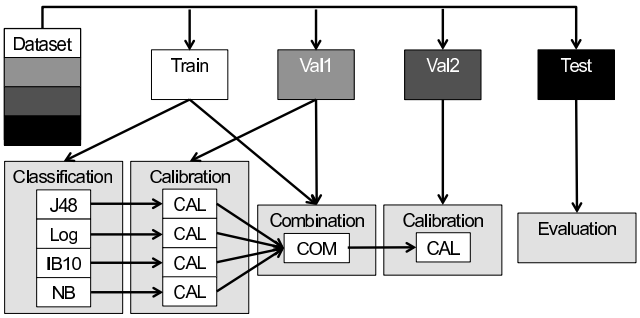

Fig. 6 Schema showing how each dataset is split into four different subsets and how they are used in each layout.

els. The other validation set (Val2) was used, in case, to calibrate the probabilities of the combined model. The training and the first validation sets were also used to tune the weights of the combination methods. The test set (Test) was used to evaluate the models. For each repetition, the same training, validation and test sets were used for all the methods.

Table 6 Datasets used in the experiments. Size, number of classes, and number of nominal and numeric attributes.

#	Datasets	Size	c	Nom.	Num.
1	Breast Cancer	286	2	9	0
2	Wisconsin Breast Cancer	699	2	0	9
3	Chess	3196	2	36	0
4	Credit Rating	690	2	9	6
5	German Credit	1000	2	13	7
6	Pima Diabetes	768	2	0	8
7	Haberman Breast	306	2	0	3
8	Heart Disease	303	2	7	6
9	Heart Statlog	270	2	0	13
10	House Voting	435	2	16	0
11	Ionosphere	351	2	0	34
12	Monks1	556	2	6	0
13	Monks2	601	2	6	0
14	Monks3	554	2	6	0
15	Mushroom	8124	2	22	0
16	Mammographic Masses	961	2	4	1
17	Sonar	208	2	0	60
18	Spam	4601	2	0	57
19	Spect	80	2	0	44
20	Tic-tac	958	2	8	0
21	Autos5c	202	5	10	15
22	Cmc	1473	3	7	2
23	Iris	158	3	0	4
24	Segmentation	2310	7	0	19
25	Tae	151	3	2	3
26	Waveform	5000	3	0	21
27	Wine	178	3	0	13
28	Vowel	990	11	3	11
29	Splice	3190	3	60	0
30	Vehicle	846	4	0	18

We evaluated the results of the different layouts for the MSE, AUC, CalBin and accuracy measures: without calibration and combination; with calibration only; with combination only; with precalibration and combination; with combination and post-calibration; and with precalibration, combination and post-calibration, as shown in Table 7. In order to ease the reproducibility of results, all the source code, scripts and datasets are available at: https://www.dropbox.com/s/f4fw1r5zi029i8b/SBA.zip.

Table 7 Experimental layouts that arrange combination and calibration.

Layout	Description and Variants
BaseModel	*BaseModel* $\in$ {J48, Logistic, IBk, NB, Random}
Base	The average of all the base models
Com	*Com* $\in$ {WCUnif, WCAcc, WCAUC, WCGINI, WCMSE, WCIMSE}
Cal	*Cal* $\in$ { PAV, Platt, Binn. }
Cal +*Com*	For different calibration and combination methods.
Com +*Cal*	For different calibration and combination methods.
Cal +*Com* +*Cal*	For different calibration and combination methods.

5.2 Experimental Results

5.2.1 Weighted Combination and Base Classifier Quality

We will first study the effect of several combination methods when there is a random classifier (i.e., a bad classifier) along with other more accurate classifiers. We are interested in a first assessment of the weighting methods in Table 2. Tables 8 and 9 show the results[2] of applying the combination methods to the four original classification models ($J48$, Log, $IB10$, and NB). The difference between Tables 8 and 9 is that Table 9 shows the results when a random classifier is added ($Random$). As a reference, we also include the average of the results of all the base classifiers ($Base$).

Clearly, when a random classifier is included, the results are worse than with only the four original classifiers. In this situation, some combination methods are more robust than others. Specifically, the $WCGINI$ and $WCIMSE$ methods obtained the best results. In order to see whether the difference for more than two methods is statistically significant, we calculated the Friedman test. When these differences are significant, we calculate the Nemenyi post-hoc test to compare all

[2] These results are averages over datasets. Although the measures for different datasets are not commensurable, we use the means to ease the presentation of results. Nonetheless, the individual results for the 30 datasets are still used for the statistical tests.

Table 8 Results for the base classifiers and different combination methods (see Table 7).

	MSE	AUC	CalBin	Acc.
J48	0.1397	0.8202	0.1062	0.7683
Log	0.1526	0.8316	0.1224	0.7752
IB10	0.1375	0.8453	0.1093	0.7590
NB	0.1487	0.8469	0.1278	0.7679
Base	0.1446	0.8360	0.1164	0.7676
WCUnif	0.1150	0.8718	0.1045	0.8052
WCAcc	0.1143	0.8724	0.1037	0.8069
WCAUC	0.1145	0.8726	0.1044	0.8065
WCGINI	0.1141	**0.8736**	0.1043	0.8077*
WCMSE	0.1145	0.8722	0.1039	0.8063
WCIMSE	**0.1109**	0.8735	**0.0960**	**0.8104**

Table 9 Results for the base classifiers and different combination methods (see Table 7), with one random classifier.

	MSE	AUC	CalBin	Acc.
J48	0.1397	0.8202	0.1062	0.7683
Log	0.1526	0.8316	0.1224	0.7752
IB10	0.1375	0.8453	0.1093	0.7590
NB	0.1487	0.8469	0.1278	0.7679
Random	0.4676	0.5009	0.4398	0.4317
Base	0.2092	0.7690	0.1811	0.7004
WCUnif	0.1298	0.8520	0.1337	0.7932
WCAcc	0.1196	0.8630	0.1175	0.8035
WCAUC	0.1203	0.8627	0.1195	0.8026
WCGINI	0.1141	0.8729	0.1048	0.8077
WCMSE	0.1208	0.8623	0.1202	0.8019
WCIMSE	**0.1117**	0.8697	**0.0986**	**0.8096**

of the methods with each other (at a confidence level of 99.5%) as suggested in [15]. The results depicted in bold in Tables 8 and 9 indicate that the method is the best and that the difference with the rest of the methods is statistically significant. The numbers that are underlined are used when more than one method is the best. More precisely, underlined numbers indicate that these results are the best and that the difference with the other methods is statistically significant, even though the difference between the underlined methods is not statistically significant. We see that $WCGINI$ and, most specially, $WCIMSE$, are the the best in many cases. $WCIMSE$ takes advantage of MSE being a metric of both separability and calibration. The good performance of these two weighting methods suggests that it is important to use an appropriate metric (AUC and MSE) instead of accuracy, but also confirms that the precise formulation of the weight is also crucial.

We also studied whether the difference between the results with and without a random classifier are statistically significant. In order to do so, we calculated the Wilcoxon Signed-Ranks test with a confidence level of 99.5% as suggested in [15]. The result shows that the difference between the pairs of methods without and with a random classifier is statistically significant, except in the case marked with the symbol $*$. The greatest differences are shown when no weighting is used ($WCUnif$). Therefore, the conclusion that comes from Tables 8 and 9 is that weights are needed when classifiers of different quality are combined, which is consistent with previous knowledge in the field [31]. However, these results show that some weighting schemes such as $WCGINI$ and $WCIMSE$ are very robust to very bad classifiers.

5.2.2 Calibration and Combination vs Combination and Calibration

The next step was to evaluate the effect of calibration and combination together. Firstly, we evaluated whether weighting was necessary when models were well calibrated. Secondly, we evaluated whether calibration was good for combination. We also wanted to know whether it was better to calibrate the base models first and combine them afterwards, or to combine the models first and to calibrate the combination afterwards.

Table 10 shows the results for each pair of calibration and combination methods[3] for the J48, Log, IB10 and NB classification models, and the random classifier[4]. We also include the average of each calibration method over the base classifiers (PAV, $Platt$, and $Binn$) and the combination by $WCGINI$ and $WCIMSE$ without calibration. We applied, again, the Friedman test to the results in Table 10 using the same notation (bold and underlined).

We can see that the results when we calibrate the model before combination (Table 10, rows 7-12) are not much better (or even worse) than an uncalibrated combination, as far as the MSE metric is concerned. The results for AUC are slightly better. In CalBin and accuracy, the difference is a little bit higher, except when Platt's calibration is used, showing that combination produces an uncalibrated classifier. When calibration is applied after combination (Table 10, rows 13-18), the results, except for the CalBin metric, are worse in general. Therefore, if we are only interested in improving the calibration of the combined models, the best option is to calibrate their probabilities after the combination. But if we want to improve the MSE, AUC and accuracy measures, it is better to first calibrate the base classifiers and then combine them.

[3] We tried the six weighting methods shown in Table 2, but the best results were obtained with $WCGINI$ and $WCIMSE$, so, in what follows, we only show these results.

[4] Apart from the magnitude in the results (values are generally better without a random classifier, as expected), the relative differences are similar, so the conclusions that can be drawn from one case (without random classifier) are similar to those that can be drawn from the other case (with a random classifier). From hereon, we will only show the results including the random classifier.

Table 10 Results for Cal, Cal+Com and Com+Cal, with one random classifier.

	MSE	AUC	CalBin	Acc.
WCGINI	0.1141	0.8729	0.1048	0.8077
WCIMSE	0.1117	0.8697	0.0986	0.8096
PAV	0.1568	0.7642	0.0966	0.7086
Platt	0.1499	0.7665	0.0982	0.7223
Binn.	0.1568	0.7610	0.0980	0.7066
PAV+WCGINI	0.1075	0.8770	0.0916	0.8171
Platt+WCGINI	0.1173	0.8779	0.1293	0.8129
Binn.+WCGINI	0.1082	0.8777	0.0945	0.8171
PAV+WCIMSE	**0.1061**	0.8753	0.0923	0.8193
Platt+WCIMSE	0.1178	0.8768	0.1342	0.8134
Binn.+WCIMSE	0.1075	0.8761	0.0980	0.8194
WCGINI+PAV	0.1141	0.8627	0.0708	0.8088
WCGINI+Platt	0.1117	0.8720	0.1029	0.8117
WCGINI+Binn.	0.1177	0.8528	0.0753	0.8033
WCIMSE+PAV	0.1137	0.8600	**0.0700**	0.8098
WCIMSE+Platt	0.1109	0.8683	0.1006	0.8129
WCIMSE+Binn.	0.1177	0.8490	0.0755	0.8030

5.2.3 Calibration before and after Combination

Finally, we are going to study the effect of Calibration + Combination + Calibration (Table 11). The idea is to check whether we can improve both the calibration and the performance of the combined model. The results show that calibrating the combined model improves the CalBin metric, but does not clearly improve MSE, accuracy and, especially, AUC. The best layout for CalBin seems to be any calibration method + WCIMSE + PAV, but only the difference between the layout PAV + WCIMSE + PAV is statistically significant compared to the rest of the results (for CalBin measure) shown in Table 11).

Table 11 MSE, AUC, CalBin and accuracy measures for Cal+Com+Cal, with one random classifier.

	MSE	AUC	CalBin	Acc.
PAV+WCGINI+PAV	0.1105	0.8680	0.0688	0.8145
PAV+WCGINI+Platt	0.1080	0.8764	0.0986	0.8176
PAV+WCGINI+Binn.	0.1147	0.8571	0.0748	0.8074
Platt+WCGINI+PAV	0.1117	0.8676	0.0699	0.8129
Platt+WCGINI+Platt	0.1091	0.8772	0.1005	0.8158
Platt+WCGINI+Binn.	0.1161	0.8553	0.0745	0.8057
Binn.+WCGINI+PAV	0.1108	0.8682	0.0685	0.8136
Binn.+WCGINI+Platt	0.1082	0.8773	0.0985	0.8171
Binn.+WCGINI+Binn.	0.1155	0.8559	0.0752	0.8058
PAV+WCIMSE+PAV	0.1093	0.8666	**0.0671**	0.8155
PAV+WCIMSE+Platt	0.1066	0.8747	0.0974	0.8194
PAV+WCIMSE+Binn.	0.1138	0.8538	0.0742	0.8091
Platt+WCIMSE+PAV	0.1098	0.8672	0.0682	0.8159
Platt+WCIMSE+Platt	0.1073	0.8757	0.0993	0.8188
Platt+WCIMSE+Binn.	0.1147	0.8536	0.0747	0.8076
Binn.+WCIMSE+PAV	0.1095	0.8671	0.0678	0.8156
Binn.+WCIMSE+Platt	0.1069	0.8751	0.0983	0.8188
Binn.+WCIMSE+Binn.	0.1144	0.8522	0.0749	0.8083

5.2.4 Summary

From the previous battery of experiments we can highlight some major findings:

- The combined model is not calibrated, as it is also shown in Section 4.
- Calibration before combination makes a limited improvement for AUC and accuracy, and no improvement (or even gets worse results) for MSE and CalBin.
- Calibration after combination gives a better picture for calibration measures, but, as expected, AUC is not increased. This is because the calibration methods are monotonic. Also, there is almost no increase in accuracy or MSE.
- Calibration + Combination + Calibration gives the best results in terms of calibration, but it is clearly an elaborate layout, which requires two validation datasets (one for each of the calibration processes).

From these results, it seems that calibration is only slightly effective for classifier combination, and weighting can do almost as well. Nonetheless, this statement should be more precise by saying that *monotonic* calibration (as given by PAV, Platt and Binning Averaging) does not bring an important push in performance for classifier combination. As a result, in the following section, we will focus on the development of a non-monotonic calibration method which tries to integrate more information from the dataset.

6 Non-monotonic Calibration

Most calibration methods are based on a univariate transformation function over the original estimated class probability, as we saw in Section 3.2. This function is always monotonic (strictly monotonic for Platt's method). One possible reason why all these methods are monotonic is because if we were allowed to modify the probabilities in a non-monotonic way, we would be prone to overfitting. In the end, calibration must be an adjustment of the estimated probability values, but not a complete change in the model properties. For calibration methods, one way of doing this is to preserve the ordering of the examples (given by their estimated probability), which can be achieved by using a monotone transformation.

However, is the previous rationale true for multiclass calibration? As we discussed in Section 3.3, PAV, binning averaging and Platt's methods are binary since they are only applied to one probability, i.e., they are univariate. Consequently, we have to use a one-vs-all or all-vs-all schema to turn binary calibration methods

into multiclass calibration methods. Nevertheless, *the extensions of binary monotonic calibration methods to multiclass calibration do not ensure monotonicity*, as the following example shows.

Example 3 Consider a classifier for a three-class problem with classes $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ which outputs the following estimations for two examples 1 and 2:

$p(1, \mathbf{a}) = 0.2, p(1, \mathbf{b}) = 0.6, p(1, \mathbf{c}) = 0.2;$

$p(2, \mathbf{a}) = 0.1, p(2, \mathbf{b}) = 0.3, p(2, \mathbf{c}) = 0.6$

After a monotonic calibration for each class, we may have the following probabilities:

$p^*(1, \mathbf{a}) = 0.7$, $p^*(1, \mathbf{b}) = 0.9$, $p^*(1, \mathbf{c}) = 0.4$;

$p^*(2, \mathbf{a}) = 0.6$, $p^*(2, \mathbf{b}) = 0.4$, $p^*(2, \mathbf{c}) = 0.5$

The rankings are maintained for the three classes, that is, $\forall\, class \in \{\mathbf{a}, \mathbf{b}, \mathbf{c}\} : p^*(i, class) > p^*(j, class)$ iff $p(i, class) > p(j, class)$. But when we normalise, we have:

$p^*(1, \mathbf{a}) = 0.35$, $p^*(1, \mathbf{b}) = 0.45$, $p^*(1, \mathbf{c}) = 0.2$;

$p^*(2, \mathbf{a}) = 0.4$, $p^*(2, \mathbf{b}) = 0.27$, $p^*(2, \mathbf{c}) = 0.33$

which breaks the monotonicity for class **a** since now $p^*(2, \mathbf{a}) > p^*(1, \mathbf{a})$ and, thus, example 2 is ranked above example 1 for class **a**.

The previous example shows that a one-vs-all approach using a monotonic calibration method does not ensure a monotonic transformation. Similar results can be obtained for the all-vs-all schema and other multiclass extensions from binary transformations simply because of the normalisation.

Therefore, does it make sense to stick to monotonic methods when, in the general multiclass case, they become non-monotonic in the end?

Following this argument, we propose the application of a calibration method which was meant to be non-monotonic from scratch [4]. The core of this approach is to change the idea of *sorting* the examples by its probability into the idea of using similarity between examples to create bins that are specific for each instance. This idea arises from the fact that if bins are created by only using the estimated probability, calibrated probabilities will be computed from possibly different examples with similar probabilities. Hence, the effect of calibration will be small since we average similar probabilities. However, if we construct the bins using similar examples according to their features, probabilities can be more diverse and calibration will have more effect.

Based on this reasoning, we have adapted a new calibration method known as Similarity-Binning Averaging (SBA) [4] for the combination setting (for which it was never analysed before). In this method the original attributes and the estimated probability are used to calculate the calibrated one.

The method is composed of two stages. The left side of Figure 7 shows Stage 1 of the SBA method. In this stage, a given model M outputs the estimated probabilities associated with a dataset. This dataset can be the same one already used for training, or an additional validation dataset VD. The estimated probabilities $p(i, j)$ $1 \leq j \leq c$ are added (as new attributes) to each instance i of VD, creating a new dataset VDP.

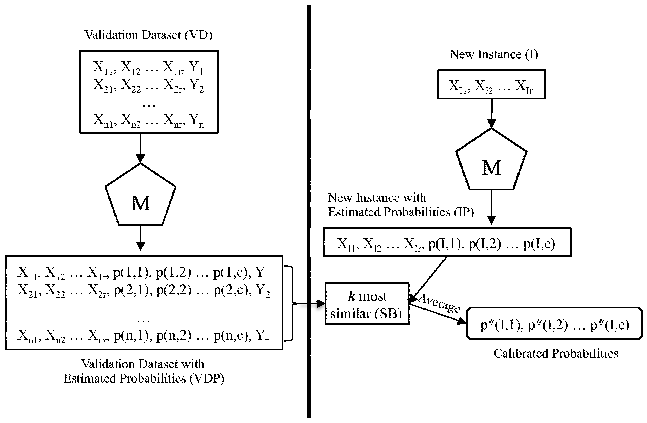

Fig. 7 Left: Stage 1 of the SBA method. Right: Stage 2 of the SBA method.

The right side of Figure 7 shows Stage 2 of the SBA method. To calibrate a new instance I, first, the estimated probability for each class is obtained from the classification model M, and these probabilities (one for each class) are added to the instance, thus creating a new instance (IP). Next, the k-most similar instances to this new instance are selected from the dataset VDP (for example, using the k-NN algorithm). This creates a bin. Finally, the calibrated probability of I for each class j is the average predicted class probability of this bin (i.e., the probability estimated by the k-NN algorithm for each class j of the instance I).

7 Experimental Results of the SBA Calibration Method

In this section, we experimentally evaluate the results of the SBA calibration method. We have seen that the calibration methods evaluated in Section 5 only produce very slight improvements (and do not improve the four studied measures equally or at the same time). So, in this section we want to evaluate whether the SBA method can change the picture in the context of classifier combination.

We will first investigate SBA as a calibration method without any combination layout, and we will

compare it with the other three calibration methods (*PAV*, *Platt* and *Binn.*). Tables 12 and 13 use the same datasets and methodology as in Section 5, but here we focus on calibration for the four calibration methods.

Table 12 MSE, AUC, CalBin and accuracy measures for base classifiers without and with calibration.

	MSE	AUC	CalBin	Acc.
J48	0.1397	0.8202	0.1062	0.7683
Log	0.1526	0.8316	0.1224	0.7752
IB10	0.1375	0.8453	0.1093	0.7590
NB	0.1487	0.8469	0.1278	0.7679
Base	0.1446	0.8360	0.1164	0.7676
PAV	0.1310	0.8301	**0.0778**	0.7779
Platt	0.1309	0.8333	0.1055	0.7749
Binn.	0.1316	0.8260	0.0807	0.7753
SBA	**0.1205**	**0.8726**	0.1022	**0.7965**

Table 13 MSE, AUC, CalBin and accuracy measures for base classifiers without and with calibration, with one random classifier.

	MSE	AUC	CalBin	Acc.
J48	0.1397	0.8202	0.1062	0.7683
Log	0.1526	0.8316	0.1224	0.7752
IB10	0.1375	0.8453	0.1093	0.7590
NB	0.1487	0.8469	0.1278	0.7679
Random	0.4676	0.5009	0.4398	0.4317
Base	0.2092	0.7690	0.1811	0.7004
PAV	0.1568	0.7642	**0.0966**	0.7086
Platt	0.1499	0.7665	0.0982	0.7223
Binn.	0.1568	0.7610	0.0979	0.7066
SBA	**0.1264**	**0.8648**	0.1080	**0.7841**

In terms of MSE, AUC and accuracy, the SBA method obtains the best results for the 20 binary datasets and the 10 non-binary datasets. In terms of CalBin, the calibration method that obtains the best results is the PAV method. The results in bold indicate that the differences are statistically significant.

Let us now investigate the effect of SBA for the combination layouts. Before analysing the experimental results, we need to point out that one possible problem of non-monotonicity is that the more transformations we do, the more overfitting may occur, and, more importantly, the correlation between the classifiers may increase (loss of diversity). In order to analyse this, we calculated Pearson's correlation coefficient for the base classifiers before and after calibrating them with the three traditional calibration techniques and SBA. While the traditional calibration techniques showed no significant increase in correlation, this was effectively higher for SBA. A higher correlation is, in theory, worse (less diversity), *unless* there is a general increase in classifier quality (when classifiers get better then they necessarily must correlate more). This latter situation seems to be more consistent here, as the results for AUC are much better. Nonetheless, a more thorough analysis on the relation between non-monotonic calibration and classifier diversity should be done. In what follows, we will focus on whether the overall results for combination are better, since many factors counter-balance here.

Next, we study the effect of Combination, Calibration + Combination, Combination + Calibration and Calibration + Combination + Calibration in Table 14 with the SBA calibration method. We compare these results with the results in Tables 8, 9, 10 and 11, using the Friedman test.

Table 14 Com, Cal+Com, Com+Cal and Cal+Com+Cal results, with one random classifier. $*$: difference non-significant with respect to PAV+WCIMSE+PAV. $\star$: difference non-significant with respect to PAV+WCIMSE. $\bullet$: difference non-significant with respect to PAV+WCIMSE and Binn.+WCIMSE.

	MSE	AUC	CalBin	Acc.
SBA+WCGINI	0.1145	0.8841	0.1124	0.8079
SBA+WCIMSE	0.1120	0.8846	0.1056	0.8104
WCGINI+SBA	0.1147	0.8781	0.0996	0.8081
WCIMSE+SBA	0.1141	0.8762	0.0975	0.8078
PAV+WCGINI+SBA	0.1123	0.8789	0.0969	0.8103
Platt+WCGINI+SBA	0.1131	0.8786	0.0978	0.8101
Binn.+WCGINI+SBA	0.1128	0.8777	0.0971	0.8086
PAV+WCIMSE+SBA	0.1100	0.8782	0.0933	0.8128
Platt+WCIMSE+SBA	0.1111	0.8781	0.0949	0.8119
Binn.+WCIMSE+SBA	0.1112	0.8765	0.0947	0.8108
SBA+WCGINI+PAV	0.1093	0.8732	0.0680	0.8161
SBA+WCGINI+Platt	0.1076	0.8842	0.1031	0.8185
SBA+WCGINI+Binn.	0.1135	0.8618	0.0734	0.8089
SBA+WCGINI+SBA	0.1149	0.8794	0.1021	0.8068
SBA+WCIMSE+PAV	0.1085	0.8732	0.0675$*$	0.8172
SBA+WCIMSE+Platt	0.1066$\star$	0.8846	0.1023	0.8198$\bullet$
SBA+WCIMSE+Binn.	0.1128	0.8620	0.0728	0.8106
SBA+WCIMSE+SBA	0.1140	0.8792	0.1002	0.8082

This table shows that SBA gives the best results in terms of AUC. The improvement is now much higher than it was for the other methods. The difference of using WCGINI or WCIMSE is not significant, but their results are still better than other weighting options (not shown in the table). The layouts with the best results in terms of AUC are SBA + WCIMSE and SBA + WCIMSE + Platt. The difference between these results and the rest of the results in Table 14 and Table 10 are statistically significant according to the Friedman test. However, in terms of the MSE metric, the difference between the result of the layout SBA + WCIMSE + Platt and PAV+WCIMSE (the best result in Table 10) is not statistically significant. In terms of CalBin, the difference between the result of the layout SBA + WCIMSE + PAV and PAV + WCIMSE + PAV (the best result in Table 11) is not statistically significant. And finally, in terms of accuracy, the difference between the result of the layout SBA + WCIMSE + Platt, PAV + WCIMSE

and Binn + WCIMSE (the best results in Table 10) is not statistically significant.

8 Discussion and Conclusions

In general terms, we now have a better understanding of classifier combination using probabilities. The separability and location of the probability distributions are the key issues in understanding classifier combination results. Measures such as AUC, MSE and CalBin are very useful in distinguishing these separability and location parameters.

Apart from all these findings, it is also worth considering whether the new weighting methods, layouts, and calibration methods are able to improve the state-of-the-art in classifier combination using probabilities. There is definitely a relevant increase in the quality of the combined models over the techniques with traditional weighting.

In order to give a clearer picture of the overall improvement, Table 15 summarises this evolution of results. We only use WCGINI because it is a weighting method that does not consider calibration, and the interpretation of results is easier. Nonetheless, very similar results are obtained for WCIMSE. For each measure we have done a significant statistical test between some of the methods. The methods which have been compared in each case are labelled by the same letter (from a to g).

Firstly, the WCUnif layout shows an unweighted combination of the base classifiers (including one random classifier). There is a clear improvement in all the parameters over the average of the base classifiers (Base) (letter a in Table 15 denotes the comparison between Base and WCUnif results). This is significantly better if we use a weighted combination using a classical combination accuracy (WCAcc) (letter b for comparing WCUnif and WCAcc results). Up to this point, this is a state-of-the-art solution. If we modify the weighting function to GINI (WCGINI), we get a significant improvement over WCAcc (letter c denotes the comparison between WCAcc and WCGINI results).

Secondly, the use of a traditional (monotonic) calibration method (Platt's) is able to improve the results in terms of AUC and accuracy both for the unweighted case (WCUnif) and for the weighted case using WCGINI (letter d in Table 15 for comparing WCUnif, WCAcc, WCGINI and Platt+WCUnif results; and letter e for comparing WCUnif, WCAcc, WCGINI and Platt+WCGINI results). Nonetheless, as discussed in previous sections: using calibration before combination typically yields better (but uncalibrated) combinations, and the improvement is not applicable to MSE or CalBin. However, this can be easily sorted out by also using a post-calibration (layout: Platt+WCGINI+Platt) (letter f for comparing WCGINI, Platt+WCUnif, Platt+WCGINI and Platt+WCGINI+Platt results).

Thirdly, the SBA calibration method is able to get further improvement, especially in terms of AUC. The layout SBA+WCGINI excels in AUC. Again, if we are interested in a calibrated combination or in good accuracy, we can use the layout SBA+WCGINI+PAV, which gives the best results in terms of MSE and CalBin (AUC is worse for this layout because PAV is not strictly monotonic and makes ties that may reduce the AUC). For accuracy, SBA+WCGINI+Platt seems a better option, while keeping AUC at its best (letter g in Table 15 for comparing WCGINI, Platt+WCGINI+Platt, SBA+WCGINI, SBA+WCGINI+SBA, SBA+WCGINI+PAV and SBA+WCGINI+Platt results).

As final recommendations, we think that classifiers that are seen as probabilistic estimators (and virtually any classifier can be converted into a probability estimator) give a more complete view of their behaviour, allowing for a more detailed combination, using their own reliabilities. The notions of diversity and quality become more complex than for crisp (non-probabilistic)

Table 15 Summary of Results (using 4 models + random classifier). For each column (MSE, AUC, CalBin, and accuracy measures) we have done a significant statistical test between the rows with the same letter. If the difference between them is significant, we have put the letter of the best result in bold; and if two or more methods are better than the rest, but the difference between them is not significant, we have underlined the letters of these methods.

	MSE	AUC	CalBin	Acc.
Base	0.2092	0.7690	0.1811	0.7004
WCUnif	0.1298	0.8520	0.1337	0.7932
WCAcc	0.1196	0.8630	0.1175	0.8035
WCGINI	0.1141	0.8729	0.1048	0.8077
Platt+WCUnif	0.1294	0.8745	0.1589	0.8063
Platt+WCGINI	0.1173	0.8779	0.1293	0.8129
Platt+WCGINI+Platt	0.1091	0.8772	0.1005	0.8158
SBA+WCGINI	0.1145	0.8841	0.1124	0.8079
SBA+WCGINI+SBA	0.1149	0.8794	0.1021	0.8068
SBA+WCGINI+PAV	0.1093	0.8732	0.0680	0.8161
SBA+WCGINI+Platt	0.1076	0.8842	0.1031	0.8185

	MSE	AUC	CalBin	Acc.
Base	a	a	a	a
WCUnif	$\mathbf{a}b\ de$	$\mathbf{a}b\ de$	$\mathbf{a}b\ de$	$\mathbf{a}b\ de$
WCAcc	$\mathbf{b}cde$	$\mathbf{b}cde$	$\mathbf{b}cde$	$\mathbf{b}cde$
WCGINI	$\mathbf{cde}fg$	$\mathbf{c}\underline{d}efg$	$\mathbf{cde}f\ g$	$\mathbf{c}\underline{d}efg$
Platt+WCUnif	$d\ f$	$\underline{d}\ f$	$d\ f$	$\underline{d}\ f$
Platt+WCGINI	ef	$\mathbf{e}\underline{f}$	ef	$\mathbf{e}f$
Platt+WCGINI+Platt	$\mathbf{f}\underline{g}$	$\underline{f}g$	$\mathbf{f}\ g$	$\mathbf{f}g$
SBA+WCGINI	g	$\underline{g}$	g	g
SBA+WCGINI+SBA	g	g	g	g
SBA+WCGINI+PAV	$\underline{g}$	g	$\mathbf{g}$	g
SBA+WCGINI+Platt	$\underline{g}$	$\underline{g}$	g	$\mathbf{g}$

classifiers, but this extra complexity can pay off with an increase in the performance of the combined model. Performance should be evaluated with several data metrics, but separability (measured in terms of AUC) is a good reference, since it is insensitive to miscalibration. Pursuing a combined model with good AUC makes sense since we know that we can calibrate a classifier with good AUC and get good accuracy results from these calibrated probabilities, using the by default thresholds (e.g. 0.5 for binary datasets).

Additionally, we have also analysed the time overload and the scalability of the several combination layouts with respect to the *Base* procedure. In Table 16 we show the time (in seconds) used in one repetition for the *Base* layout for ten different datasets (first row). Concretely, we use three datasets with two classes (14, 3 and 15 in Table 6) (one of the smallest dataset, one medium dataset and the biggest dataset), three datasets with three classes (23, 22 and 26 in Table 6) (one of the smallest dataset, one medium dataset and the biggest dataset) and four datasets with four, five, seven and eleven classes respectively (30, 21, 24 and 28 in Table 6). The rest of the table shows the relative increment in time (percentage) of the other layouts with respect to the *Base* layout. The table shows that there is of course a time overload for the layouts with calibration, but it is still in reasonable figures for medium-sized problems. We see that as far as the datasets become larger and more complex, the relative overhead (over a single classifier) increases slightly with respect to simple datasets. Regarding the calibration methods used, binning is the most efficient, while Platts shows some scalability problems for some datasets. The SBA method somewhat lies in between.

Summing up, from all these results and analyses, we would like to highlight some clear messages, as follows:

- Calibration is beneficial before combination as the experimental results show, *in general*. Monotonic calibration methods have a more limited influence than non-monotonic ones.
- The combination of classifiers does not typically give a calibrated result, as we have shown by analysing the probability distributions using truncated normal models for them. This has been confirmed by the experimental results.
- We advocate for AUC as the right measure to evaluate combination performance, precisely because the combination is generally uncalibrated.
- We recommend calibration after combination, if we are interested in good results in terms of MSE or in terms of accuracy.
- Weighted combination is compatible with probabilities even when we use calibration with the same dataset from which we derive the weights. This has been shown by the experiments. Therefore, the *double-weighting* phenomenon is not really a problem, or at least it is counteracted by other benefits.
- The weighting methods which are best when using probabilities are GINI and IMSE, even in conjunction with calibration.
- SBA, the non-monotonic calibration method, is better for combination according to the experimental results.

This better understanding of classifier combination using probabilities is not only useful for the general case, but for specific applications and problems. We now have tools to analyse how classifiers change with calibration and combination. The distribution plots we used in Section 4 can be used to analyse the results of calibration and combination of a specific set of classifiers. The use of several metrics (such as AUC, MSE, accuracy, and CalBin) are a requirement to understand what is really going on when classifiers are transformed and combined.

Finally, we have also raised many new questions. More elaborate tools could be used to analyse probability distributions theoretically, especially to address the general multiclass case. The use of other diversity measures, such as Spearman's rank correlation would also be insightful. Empirical results can also be extended with more layouts, different settings, datasets sizes and features, model types, etc. In the end, calibration is a complex phenomenon by itself, which becomes even more convoluted when coupled with the already multifarious area of model combination.

Acknowledgements We thank the anonymous reviewers for their comments, which have helped to improve this paper significantly. This work was supported by the MEC/MINECO projects CONSOLIDER-INGENIO CSD2007-00022, COST action IC0801 and TIN 2010-21062-C02-02, GVA project PROMETEO/2008/051, and the REFRAME project granted by the European Coordinated Research on Long-term Challenges in Information and Communication Sciences & Technologies ERA-Net (CHIST-ERA), and funded by the Ministerio de Economía y Competitividad in Spain.

References

1. Amemiya, T.: Regression Analysis when the Dependent Variable Is Truncated Normal. Econometrica **41**(6), 997–1016 (1973)
2. Ayer, M., Brunk, H., Ewing, G., Reid, W., Silverman, E.: An empirical distribution function for sampling with incomplete information. Annals of Mathematical Statistics **5**, 641–647 (1955)
3. Bella, A., Ferri, C., Hernandez-Orallo, J., Ramirez-Quintana, M.: Calibration of machine learning models.

In: Handbook of Research on Machine Learning Applications, pp. 128–146. IGI Global (2009)
4. Bella, A., Ferri, C., Hernández-Orallo, J., Ramírez-Quintana, M.: Similarity-Binning Averaging: A Generalisation of Binning Calibration. In: Intelligent Data Engineering and Automated Learning - IDEAL 2009, *Lecture Notes in Computer Science*, vol. 5788, pp. 341–349. Springer Berlin / Heidelberg (2009)
5. Bennett, P.N.: Building reliable metaclassifiers for text learning. Ph.D. thesis, Carnegie Mellon University (2006)
6. Bennett, P.N., Dumais, S.T., Horvitz, E.: The Combination of Text Classifiers Using Reliability Indicators. Information Retrieval **8**(1), 67–98 (2005)
7. Blake, C., Merz, C.: UCI repository of machine learning databases (1998). URL http://www.ics.uci.edu/∼mlearn/MLRepository.html
8. Breiman, L.: Bagging predictors. Machine Learning **24**, 123–140 (1996)
9. Brier, G.: Verification of forecasts expressed in terms of probabilities. Monthly Weather Review **78**, 1–3 (1950)
10. Brümmer, N.: Measuring, refining and calibrating speaker and language information extracted from speech. Ph.D. thesis, University of Stellenbosch (2010)
11. Canuto, A., Santos, A., Vargas, R.: Ensembles of artmap-based neural networks: an experimental study. Applied Intelligence **35**, 1–17 (2011)
12. Caruana, R., Munson, A., Mizil, A.N.: Getting the Most Out of Ensemble Selection. In: ICDM '06: Proceedings of the Sixth International Conference on Data Mining, pp. 828–833. IEEE Computer Society, Washington, DC, USA (2006)
13. Caruana, R., Niculescu-Mizil, A.: Data mining in metric space: an empirical analysis of supervised learning performance criteria. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '04, pp. 69–78. ACM, New York, NY, USA (2004)
14. Cohen, I., Goldszmidt, M.: Properties and benefits of calibrated classifiers. In: Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD '04, pp. 125–136. Springer-Verlag (2004)
15. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research **7**, 1–30 (2006)
16. Dietterich, T.G.: Ensemble methods in machine learning. In: Proceedings of the First International Workshop on Multiple Classifier Systems, MCS '00, pp. 1–15. Springer-Verlag, London, UK (2000)
17. Dietterich, T.G.: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning **40**, 139–157 (2000)
18. Fahim, M., Fatima, I., Lee, S., Lee, Y.: Eem: evolutionary ensembles model for activity recognition in smart homes. Applied Intelligence pp. 1–11 (2012)
19. Ferri, C., Flach, P., Hernández-Orallo, J.: Delegating classifiers. In: Proceedings of the twenty-first international conference on Machine learning, ICML '04, pp. 37–45. ACM, New York, NY, USA (2004)
20. Ferri, C., Hernández-Orallo, J., Modroiu, R.: An experimental comparison of performance measures for classification. Pattern Recognition Letters **30**, 27–38 (2009)
21. Ferri, C., Hernández-orallo, J., Salido, M.: Volume under the roc surface for multi-class problems. Exact computation and evaluation of approximations. In: Proceedings of 14th European Conference on Machine Learning, pp. 108–120 (2003)
22. Flach, P., Blockeel, H., Ferri, C., Hernández-Orallo, J., Struyf, J.: Decision support for data mining: An introduction to ROC analysis and its applications. In: Data Mining and Decision Support: Integration and Collaboration, pp. 81–90. Kluwer Academic Publishers, Boston (2003)
23. Freund, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm. In: International Conference on Machine Learning, pp. 148–156 (1996)
24. Gama, J., Brazdil, P.: Cascade generalization. Machine Learning **41**, 315–343 (2000)
25. Garczarek, U.: Classification rules in standardized partition spaces. Ph.D. thesis, Universitat Dortmund (2002)
26. Gebel, M.: Multivariate calibration of classifier scores into the probability space. Ph.D. thesis, University of Dortmund (2009)
27. Hand, D.J., Till, R.J.: A simple generalisation of the area under the roc curve for multiple class classification problems. Machine Learning **45**, 171–186 (2001)
28. Hoeting, J.A., Madigan, D., Raftery, A.E., Volinsky, C.T.: Bayesian model averaging: A tutorial. Statistical Science **14**(4), 382–417 (1999)
29. Khor, K., Ting, C., Phon-Amnuaisuk, S.: A cascaded classifier approach for improving detection rates on rare attack categories in network intrusion detection. Applied Intelligence **36**, 320–329 (2012)
30. Kuncheva, L.I.: A theoretical study on six classifier fusion strategies. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**, 281–286 (2002)
31. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)
32. Kuncheva, L.I.: Diversity in multiple classifier systems. Information Fusion **6**(1), 3 – 4 (2005). Diversity in Multiple Classifier Systems
33. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning **51**, 181–207 (2003)
34. Lee, H., Kim, E., Pedrycz, W.: A new selective neural network ensemble with negative correlation. Applied Intelligence pp. 1–11 (2012)
35. Maudes, J., Rodríguez, J., García-Osorio, C., Pardo, C.: Random projections for linear svm ensembles. Applied Intelligence **34**, 347–359 (2011)
36. Murphy, A.H.: Scalar and vector partitions of the probability score: Part II. n-state situation. Journal of Applied Meteorology **11**, 1182–1192 (1972)
37. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers, pp. 61–74. MIT Press, Boston (1999)
38. Raftery, A.E., Gneiting, T., Balabdaoui, F., Polakowski, M.: Using bayesian model averaging to calibrate forecast ensembles. monthly weather review 133 (2005)
39. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. Journal of Machine Learning Research **5**, 101–141 (2004)
40. Robertson, T., Wright, F.T., Dykstra, R.L.: Order Restricted Statistical Inference. John Wiley & Sons (1988)
41. Souza, L., Pozo, A., Rosa, J., Neto, A.: Applying correlation to enhance boosting technique using genetic programming as base learner. Applied Intelligence **33**, 291–301 (2010)
42. Tulyakov, S., Jaeger, S., Govindaraju, V., Doermann, D.: Review of classifier combination methods. In: H.F. Simone Marinai (ed.) Studies in Computational Intelligence: Machine Learning in Document Analysis and Recognition, pp. 361–386. Springer (2008)

43. Verma, B., Hassan, S.: Hybrid ensemble approach for classification. Applied Intelligence **34**, 258–278 (2011)
44. Wang, C., Hunter, A.: A low variance error boosting algorithm. Applied Intelligence **33**, 357–369 (2010)
45. Witten, I.H., Frank, E.: Data mining: practical machine learning tools and techniques with java implementations. SIGMOD Record **31**, 76–77 (2002)
46. Wolpert, D.H.: Stacked generalization. Neural Networks **5**, 241–259 (1992)
47. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '02, pp. 694–699. ACM, New York, NY, USA (2002)

Table 16 The row *Base* shows the time (in seconds) for the *Base* layout for ten different datasets. The other rows show the relative increment in time (percentage) of the other layouts with respect to the *Base* layout.

	14	3	15	23	22	26	30	21	24	28
Base	0.3	3.5	7.9	0.2	0.6	9.1	0.4	0.7	1.1	0.6
PAV	300%	454%	1051%	200%	717%	774%	525%	229%	791%	833%
Platt	1033%	403%	497%	850%	2583%	656%	2825%	714%	4927%	6033%
Binn.	233%	214%	214%	200%	217%	211%	250%	214%	227%	233%
SBA	467%	663%	678%	400%	533%	829%	600%	400%	691%	500%
WCGINI	233%	291%	335%	200%	283%	330%	300%	214%	336%	300%
PAV+WCGINI	433%	857%	2065%	300%	1383%	1538%	1000%	357%	1582%	1667%
Platt+WCGINI	2000%	780%	1015%	1600%	5100%	1332%	5600%	1314%	9909%	12133%
Binn.+WCGINI	400%	391%	418%	300%	383%	421%	450%	329%	464%	467%
SBA+WCGINI	1033%	1720%	1862%	800%	1300%	2219%	1500%	829%	1964%	1350%
WCGINI+PAV	433%	451%	771%	300%	633%	710%	575%	329%	545%	767%
WCGINI+Platt	567%	534%	524%	500%	967%	563%	1100%	443%	1618%	1867%
WCGINI+Binn.	367%	389%	437%	300%	367%	431%	425%	343%	445%	417%
WCGINI+SBA	433%	500%	546%	350%	450%	580%	500%	357%	555%	467%
PAV+WCGINI+PAV	600%	1197%	3015%	400%	2267%	2251%	1525%	486%	2364%	2667%
PAV+WCGINI+Platt	800%	1334%	3094%	600%	2400%	2232%	2075%	586%	3473%	3900%
PAV+WCGINI+Binn.	600%	1186%	3004%	450%	2000%	2143%	1400%	471%	2300%	2450%
PAV+WCGINI+SBA	700%	1297%	3120%	500%	2067%	2295%	1500%	529%	2409%	2500%
Platt+WCGINI+PAV	2933%	1163%	1663%	2400%	7867%	2158%	8450%	1886%	14918%	18433%
Platt+WCGINI+Platt	3033%	1154%	1508%	2550%	8167%	1970%	8975%	2014%	15927%	19500%
Platt+WCGINI+Binn.	2867%	1080%	1418%	2350%	7600%	1882%	8300%	1914%	14764%	18083%
Platt+WCGINI+SBA	2933%	1189%	1528%	2400%	7667%	2031%	8375%	1929%	14873%	18200%
Binn.+WCGINI+PAV	533%	506%	542%	400%	783%	641%	725%	457%	718%	900%
Binn.+WCGINI+Platt	700%	643%	551%	650%	917%	620%	1300%	586%	1791%	2083%
Binn.+WCGINI+Binn.	500%	497%	533%	400%	517%	531%	600%	443%	618%	650%
Binn.+WCGINI+SBA	600%	609%	644%	450%	600%	681%	675%	500%	727%	700%
SBA+WCGINI+PAV	1567%	2446%	2982%	1200%	2083%	3312%	2250%	1229%	2855%	2167%
SBA+WCGINI+Platt	1667%	2451%	2629%	1450%	2433%	3134%	2800%	1343%	3855%	3333%
SBA+WCGINI+Binn.	1533%	2383%	2543%	1200%	1833%	3042%	2125%	1214%	2673%	1883%
SBA+WCGINI+SBA	1567%	2491%	2651%	1250%	1900%	3196%	2200%	1286%	2773%	1950%