Calibrating Multi-Class Models

Ulf Johansson ULF.JOHANSSON@JU.SE
Dept. of Computing, Jönköping University, Sweden

Tuwe Löfström TUWE.LOFSTROM@JU.SE
Dept. of Computing, Jönköping University, Sweden

Henrik Boström BOSTROMH@KTH.SE
School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden

Editor: Lars Carlsson, Zhiyuan Luo, Giovanni Cherubin and Khuong An Nguyen

Abstract

Predictive models communicating algorithmic confidence are very informative, but only if well-calibrated and sharp, i.e., providing accurate probability estimates adjusted for each instance. While almost all machine learning algorithms are able to produce probability estimates, these are often poorly calibrated, thus requiring external calibration. For multi-class problems, external calibration has typically been done using one-vs-all or all-vs-all schemes, thus adding to the computational complexity, but also making it impossible to analyze and inspect the predictive models. In this paper, we suggest a novel approach for calibrating inherently multi-class models. Instead of providing a probability distribution over all labels, the estimation is of the probability that the class label predicted by the underlying model is correct. In an extensive empirical study, it is shown that the suggested approach, when applied to both Platt scaling and Venn-Abers, is able to improve the probability estimates from decision trees, random forests and extreme gradient boosting.

Keywords: Multi-class, Calibration, Probabilistic classifiers, Platt scaling, Venn-Abers

1. Introduction

Classifiers accompanying the predicted label with some measure of *algorithmic confidence* are expected to increase a user's appropriate trust. The most obvious example of such models is *probabilistic predictors* that, for each prediction, return a probability distribution over all possible labels. If these models are both *well-calibrated*, and *sharp*, a user will get probability estimates reflecting the true underlying probabilities on the instance level.

While almost all machine learning algorithms are capable of producing some measures of confidence, these can rarely be interpreted as probabilities. With this in mind, a number of external calibration methods have been suggested, converting these internal confidences into probability estimates. The two most well-known calibration techniques are Platt scaling (Platt, 1999) and isotonic regression (Zadrozny and Elkan, 2001). Both these techniques fit a function (either a logistic or an isotonic) to the confidence measures and the true targets on a calibration set. Fitting a function, however, requires that the underlying model is a *scoring classifier*, i.e., that higher confidences indicate a larger belief in the positive class.

As a consequence, these calibration techniques are only directly applicable to two-class problems.

Another technique that can be used for calibration is so-called *Venn-Abers predictors* (Vovk and Petej, 2012). Venn-Abers also operates on scoring classifiers, producing multi-probabilistic predictors with unique validity properties. While these multi-probabilistic predictors are highly informative, and could be used for both prediction and analysis, they are also calibrators similar to Platt scaling and isotonic regression, see e.g., (Johansson et al., 2019).

For multi-class problems, i.e., when there are more than two classes but each instance should be given exactly one class label, the standard approach has been to use either one-vs-all or all-vs-all schemes, before performing calibration for each class and then aggregating the results into probability estimates. In addition to the obvious drawback of having to train a number of models, it must be noted that when using these approaches, there is no longer one predictive model used for the predictions, but a set of models. One consequence of this is that even if the algorithm employed produces interpretable models, like decision trees or rule sets, it is no longer possible to inspect and analyze a single model to understand the logic behind the predictions.

In this study, we suggest a novel way of calibrating multi-class models, restricting the investigation to *inherently multi-class* models, i.e., models that are capable of predicting not only a label, but also a confidence measure for each possible label. For such models, an obvious approach to creating probabilistic predictors, without using one-vs-all or all-vs-all schemes, would simply be to directly apply the inherently multi-class model to a calibration set. Here, it must be noted that we make one natural, but also very important, simplification; the predicted label may not change due to the calibration. Instead, the output will always be the same label as predicted by the model, and we are only interested in the probability estimate for that label. This is of course very different from most calibration techniques that output probabilities for all class labels, and where the predicted label may very well change due to the calibration. One straightforward interpretation of the output probability estimates of the proposed approach is that they provide an estimate of whether the prediction is correct or not. With this setup, the key idea is to for each prediction consider the predicted label as the positive class, and all other labels as the negative class, thus making it possible to utilize the calibration techniques requiring scoring classifies.

In the empirical investigation, we use three different inherently multi-class techniques; decision trees, random forests and extreme gradient boosting (xGB) as underlying models. For the calibration, we compare uncalibrated models to models calibrated using the suggested approach together with either Platt scaling or Venn-Abers for the actual calibration.

In the next section, we describe probabilistic prediction, Platt scaling, Venn-Abers and multi-class calibration, before outlining the algorithms producing the underlying models in the experimentation. In Section 3, we introduce the suggested approach and describe the experimental setup, including the data sets used. In Section 4, we first demonstrate the approach and then present and analyze the results obtained when using each of the three types of underlying models. Finally, in Section 5, we give the main conclusions and outline some directions for future work.

2. Background

2.1. Probabilistic prediction

As described above, a probabilistic predictor outputs both the predicted class label and a probability distribution over the labels. If the predicted probability distributions perform well against statistical tests based on subsequent observation of the labels, they are considered *valid*. However, as shown by Gammerman et al. (1998), validity can not, in a general sense, be achieved for probabilistic prediction. In this paper, we focus, however, only on *calibration*:

$$p(c_j \mid p^{c_j}) = p^{c_j}, \tag{1}$$

where p^{c_j} is the probability estimate for class j. In order to not be misleading, the predicted probabilities must be matched by observed accuracy. In other words, if a number of predictions with the probability estimate 0.9 are made, these predictions should be correct in about 90% of the cases. While most predictive models are inherently capable of producing probability estimates, the achieved estimates are often poorly calibrated. The typical way of handling poorly calibrated models is to apply some external calibration method, where Platt scaling (Platt, 1999) and isotonic regression (Zadrozny and Elkan, 2001) are the two most frequently used. The standard procedure for these external methods is to perform the actual calibration on a separate part of the available labelled data called the *calibration set*.

2.2. Platt scaling

Platt scaling (Platt, 1999) fits a sigmoid function to the scores obtained by the model on the calibration set. Given a probability estimation score s, the function is

$$\hat{p}(c \mid s) = \frac{1}{1 + e^{As+B}}, \tag{2}$$

where $\hat{p}(c \mid s)$ is the probability that an instance belongs to class c, given its score s. The parameters A and B are found by a gradient descent search, minimizing the loss function as suggested by Platt (1999).

To avoid the risk of infinite losses due to the estimates being exactly 0 or 1, regularization is often applied. The regularization use the following target values (where k_+ and k_- are the number of calibration instances labeled 1 and 0, respectively) instead of 0 and 1:

$$\begin{aligned} t_+ &:= \frac{k_+ + 1}{k_+ + 2} \\ t_- &:= \frac{1}{k_- + 2} \end{aligned} \tag{3}$$

2.3. Venn-Abers predictors

Venn predictors introduced by Vovk et al. (2004) are probabilistic predictors that circumvent the general impossibility result regarding validity by (i) restricting the statistical tests for validity to calibration, and (ii) outputting multiple probabilities for each label, with one of them being the valid one. As described below, these multiprobabilistic predictions can be converted into probability intervals for each label, where the size of the intervals gives a

crude indication of the confidence in the estimation. Inductive Venn prediction (Lambrou et al., 2015) uses an underlying model to divide the calibration instances into a number of *categories*, based on a so-called *Venn taxonomy*. For each class label, the relative frequency of calibration instances with that label and falling into a category is then used as the estimated probability for test instances falling into the same category. Validity is achieved by including the test instance in this calculation. To handle the fact that the true label is unknown for test instances, every possible label is tried, and the resulting probability distribution is calculated. This results in a set of C label probability distributions, where C is the number of possible labels. For an extended introduction to Venn predictors, see (Johansson et al., 2019).

A critical decision when using Venn predictors is to pick a suitable taxonomy. However, for *scoring classifiers*, described below, *Venn-Abers predictors* (Vovk and Petej, 2012) offer an alternative where the taxonomy is automatically optimized using isotonic regression. Since Venn-Abers predictors are Venn predictors, they inherit the validity guarantees.

A scoring classifier is a classifier restricted to two-class problems that, when making a prediction for a test object x_i, outputs a *prediction score* $s(x_i)$, where a higher value indicates a larger belief in label 1. To obtain the predicted class label from a scoring classifier, the score is compared to a fixed threshold t, and the prediction is 1 if $s(x) > t$, and otherwise 0. A Venn-Abers predictor requires a scoring classifier as the underlying model. Instead of using a fixed threshold, an increasing function g is fitted using a number of prediction scores with known true targets. This function, $g(s(x))$, can then be interpreted as the probability that the label for x is 1, i.e., it is a calibrator. Venn-Abers predictors use isotonic regression (Zadrozny and Elkan, 2001) for the fitting.

An isotonic calibrator is a step-wise, non-decreasing, regression function typically produced by the pair-adjacent violators algorithm. The algorithm starts with a set of input probability intervals, where the borders are the scores of the calibration instances. It then repeatedly merges adjacent intervals where the lower interval contains a higher (or equally high) fraction of examples belonging to the positive class. This process continues until no such pair of intervals can be found. When the algorithm has terminated, it outputs a function that, for each interval, returns the fraction of positive examples in the calibration set in that interval.

A multiprobabilistic prediction from an inductive Venn-Abers predictor is produced as follows:

1. Let $\{z_1, \ldots, z_{l+q}\}$ be a training set where each instance $z_i = (x_i, y_i)$ consists of two parts; an *object* x_i and a *label* y_i.

2. Let the training set be divided into a proper training set Z_T with q instances and a calibration set $\{z_1, \ldots, z_l\}$.

3. Train a scoring classifier using the proper training set Z_T to produce the prediction scores s_0 for $\{z_1, \ldots, z_l, (x_{l+1}, 0)\}$ and s_1 for $\{z_1, \ldots, z_l, (x_{l+1}, 1)\}$.

4. Let g_0 be the isotonic calibrator for $\{(s_0(x_1), y_1), \ldots, (s_0(x_l), y_l), (s_0(x_{l+1}), 0)\}$ and g_1 be the isotonic calibrator for $\{(s_1(x_1), y_1), \ldots, (s_1(x_l), y_l), (s_1(x_{l+1}), 1)\}$.

5. Let the probability interval for $y_{l+1} = 1$ be $[g_0(s_0(x_{l+1})), g_1(s_1(x_{l+1}))]$.

2.4. Multi-class calibration

For multi-class calibration, Zadrozny and Elkan (2002) suggested a one-vs-all approach, training a binary classifier on each split before calibrating. In general, such an approach is applicable to any underlying model, and to using any binary calibration technique. Exactly how to combine the calibrated estimates is not obvious though.

When Manokhin (2017) used standard and cross Venn-Abers for multi-class calibration, a pair-wise (all-vs-all) approach was suggested and applied on models built by logistic regression, support vector machines and neural networks. The overall result was that probabilistic models calibrated with Venn-Abers were generally better calibrated than both the uncalibrated predictors and when using Platt scaling.

As argued by Wenger et al. (2020), however, most modern classifiers are inherently multi-class, thus offering an alternative to the traditional approach with one-vs-all or all-vs-all schemes. More specifically, instead of training a number of models before calibrating and aggregating, the predictions from the inherently multi-class model on a calibration set can be used directly for calibration.

2.5. Probability Estimation Trees

Decision trees are interpretable models that are rather accurate and fast to train with relatively few parameters to tune. The decision tree algorithm uses a greedy divide-and-conquer strategy which iteratively splits the training data into smaller subsets using the split that maximizes the gain as determined by a loss function. Every split results in an internal node, using the split as a condition. Each subset is recursively split into smaller and smaller parts until it is not possible to find any split that increase the gain. When no more splits can be found, leaf nodes are created, using the most frequent class label in the leaf as the prediction.

A decision tree producing probabilistic predictions is called a probability estimation tree (PET) (Provost and Domingos, 2003). The most straightforward probability estimate for a decision tree is to use the relative frequency of the classes in the leaf nodes. This is also what is used by the *DecisionTreeClassifier* in *scikit-learn*.

2.6. Random forests

Random forests (Breiman, 2001) are ensembles consisting of random trees. A random tree is a decision tree trained i) using bagging, and ii) by only considering a random subset of the available attributes when choosing each split. When using bagging, a bootstrap replicate is drawn by sampling q instances, with replacement, from q training instances.

When using the *RandomForestClassifier* in *scikit-learn*, the class probability of a single tree is the fraction of samples of the same class in a leaf. In order to get the probability estimates of the forest, the mean predicted class probabilities of all trees is calculated.

2.7. Extreme Gradient Boosting (xGB)

Extreme Gradient Boosting (Chen and Guestrin, 2016), often called *XGBoost*, is a highly scalable implementation of gradient boosting (Friedman, 2001). A gradient boosting ensemble is built sequentially, one tree at a time. At each iteration, the current ensemble

predicts the training data. The gradients for each instance are calculated according to the loss function and the gradients are used to determine gradient histograms for each attribute and every leaf in the tree. The gradient histograms determine the gradient distribution, for each combination of attribute and leaf and for every possible value of an attribute. The gradient distribution of the right and left child of the parent leaf determines if an attribute would be used as splitting criteria. The optimal split point, maximizing the gain as determined by the loss function, is used to grow the tree.

XGBoost includes several improvements over the original gradient boosting algorithm, intended to enable parallel and distributed computation of sparse data, and weighted quantile sketch for approximate tree learning, for details see, (Chen and Guestrin, 2016).

The *XGBClassifier* in the *xgboost* package, implementing the algorithm in Python, uses the same calculation to get the probability estimate as the *RandomForestClassifier*.

3. Method

In the suggested approach, an underlying model is first trained using a proper training set, before it is applied to the calibration set, resulting in a set of confidence measures for each label. The procedure then utilizes the same underlying model and calibration set for the calibration of all test instances.

More specifically, to produce a probability estimate for a test instance, the label predicted by the underlying model is considered the positive class, and all other labels are regarded as belonging to the negative class. After this, the actual calibration is performed in the standard way for Platt scaling and Venn-Abers, resulting in a probability estimate for the positive class, i.e., the label predicted.

It may be noted that for Platt scaling, the fitting is only performed once for each label, but Venn-Abers requires two isotonic regressions for each test instance. The standard approach, of course, trains at least C models, where C is the number of classes.

When comparing Venn-Abers calibrations to other techniques, the output probability intervals (p_0, p_1), must be aggregated into a single probability estimate. In this study, we follow the recommendation from Vovk and Petej (2012) and use a regularized value:

$$p = \frac{p_1}{1 - p_0 + p_1} \tag{4}$$

In summary, we compare the following three setups:

- No external calibration (NoCal): Uses the output from the underlying model as probability estimates. Since this setup requires no calibration set, all available labelled data were used for inducing the models.

- Platt scaling (Platt): One logistic function is fitted to the calibration set for each label, considering all other labels as the negative class. For each test instance, the label predicted by the underlying model is the positive class.

- Venn-Abers (VA): For each test instance, the label predicted by the underlying model is considered the positive class, and then two isotonic regressions are fitted to the calibration set; once augmented with the test instance labelled as the positive class,

and once augmented with the test instance labelled as the negative class. The resulting probability interval is then converted into a single probability estimate for the positive class using eq. 4.

3.1. Experimental setup

In the experimentation, *scikit-learn*, was used and all parameter values were left at the default settings with one exception. For the decision trees, the parameter *min_samples_leaf* was set to 4, i.e., each leaf should contain at least four training instances. For the evaluation, standard 10x10-fold stratified cross-validation was used. For Platt scaling and Venn-Abers, the proper training set consisted of 2/3 of all the training instances and the calibration set of 1/3. For the non-calibrated models, all training data was, as mentioned above, used for generating the model. In the experiments, 20 publicly available multi-class data sets from the UCI repository (Dua and Graff, 2017) were used. The data sets characteristics are presented in Table 1, where *#class* is the number of classes, *#inst.* is the number of instances and *#attrib.* is the number of input attributes.

Table 1: Data sets

Data set	#class	#inst.	#attrib.	Data set	#class	#inst.	#attrib.
balance	3	625	4	tae	3	151	5
cars	4	1728	6	user	5	403	5
cmc	3	1473	9	wave	3	5000	40
cool	3	768	8	vehicle	4	846	18
ecoli	8	336	7	whole	3	440	7
glass	6	214	9	wine	3	178	13
heat	3	768	8	wineR	6	1599	11
image	7	2310	19	wineW	7	4898	11
iris	3	150	4	vowel	11	990	11
steel	7	1941	27	yeast	10	1484	8

For the evaluation, accuracy and area under the ROC-curve (AUC) are used to measure the predictive performance. Investigating the quality of the calibration, we report log losses and the expected calibration error (ECE). The log loss is calculated using

$$\lambda_{log} = \begin{cases} -\log p & \text{if correct} \\ -\log(1-p) & \text{if incorrect} \end{cases} \tag{5}$$

where log is the binary logarithm and p the estimate for the predicted label. It should be noted that the log loss function used (from scikit-learn) avoids infinite results by clipping the probabilities making sure that they never are exactly 0 or 1.

When calculating ECE, the probability estimates for the predicted class are divided into M (here $M = 10$) equally sized bins, before taking a weighted average of the absolute differences between the fraction of correct (foc) predictions and the mean of the prediction probabilities (mop), see Eq. 6 where n is the size of the data set and B_i represents bin i.

$$ECE = \sum_{i=1}^{M} \frac{|B_i|}{n} \Big| foc(B_i) - mop(B_i) \Big| \tag{6}$$

4. Results

In this section, we first demonstrate the suggested approach before presenting the results for each of the three underlying model types.

4.1. Demonstration

Fig. 1 shows an induced VA-tree for the Image data set. The tree parameter *min_weight_fraction_leaf* was here set to 0.1 to force the tree to be small enough for this analysis. Each leaf, with corresponding intervals for the calibrated probabilities, predict one of the seven classes. The sizes of the intervals are dependent on the number of instances falling into the leaves, with more data resulting in smaller intervals. As a consequence of the small tree, with a large number of instances in each leaf, all intervals will be fairly tight. In a fully-grown tree, the interval sizes could be expected to vary more, as a consequence of the much larger variation in the number of instances falling into each leaf node. In the long run, we would expect the true error rate in each leaf node to be within the interval as a consequence of the probabilities being well-calibrated. In this example, we would expect to be correct $66.7 - 68.0$ % of the time when predicting *window*, whereas we would be almost certain when predicting *sky* or *grass*. In fact, both these leaves are 100 % accurate in our example.

```
exgreen-mean <= 101.778
|   saturatoin-mean <= 0.892
|   |   region-centroid-col <= 158.500
|   |   |   intensity-mean <= 27.389
|   |   |   |   saturatoin-mean <= -1.876
|   |   |   |   |   saturatoin-mean <= -2.151
|   |   |   |   |   |    P(0.705, 0.718) class: foliage
|   |   |   |   |   saturatoin-mean >  -2.151
|   |   |   |   |   |    P(0.667, 0.680) class: window
|   |   |   |   saturatoin-mean >  -1.876
|   |   |   |   |    P(0.791, 0.802) class: brickface
|   |   |   intensity-mean >  27.389
|   |   |   |    P(0.893, 0.911) class: cement
|   |   region-centroid-col >  158.500
|   |   |    P(0.918, 0.932) class: path
|   saturatoin-mean >  0.892
|   |    P(0.993, 1.000) class: grass
exgreen-mean >  101.778
|    P(0.993, 1.000) class: sky
```

Figure 1: Venn-Abers calibrated tree for the Image data set

Table 2 lists a few examples of instances predicted by random forest and XGBoost. A decision-maker using the predictions from our proposed solution would use the information in the prediction and probability columns. The target column provides the ground truth for these instances as comparison. As can be expected, predictions with higher probabilities are more likely to be correct. A direct consequence of the connection between interval size and the amount of data is that smaller data sets generally have larger intervals, as can be seen when comparing the *tae* (with only 151 instances) and the *cmc* (with 1473 instances) data

sets. Despite very similar accuracy, *tae* have much wider intervals due to fewer instances. The probabilities over an entire data set can be expected to average to values close to the accuracy achieved on the data set. However, as can be seen, probabilities can range from intervals including very low probabilities (rows 4, 5, and 11), to intervals including very high probabilities (rows 2, 3, and 8).

Row	Algorithm	Data set	Target	Prediction	Probability
1	RF	tae	Low	Low	P(0.40, 0.60)
2	RF	tae	High	High	P(0.83, 1.00)
3	RF	tae	High	Medium	P(0.50, 1.00)
4	xGB	tae	Medium	High	P(0.00, 0.21)
5	xGB	tae	Low	Low	P(0.14, 0.33)
6	xGB	vehicle	Opel	Saab	P(0.66, 0.67)
7	xGB	vehicle	Bus	Bus	P(0.75, 0.86)
8	xGB	vehicle	Van	Van	P(0.97, 1.00)
9	RF	cmc	No-use	No-use	P(0.45, 0.46)
10	RF	cmc	Short-term	Short-term	P(0.65, 0.66)
11	RF	cmc	Short-term	No-use	P(0.14, 0.25)

Table 2: Examples of predicted instances with the VA-calibrated intervals

4.2. Results for decision trees

Starting with the predictive performance, we see that using external calibration usually results in lower accuracy and AUC. The reason is of course the need for a separate calibration set reducing the labelled data available for the training of the model. Here it should be noted, as described above, that in this setting both Platt scaling and Venn-Abers will predict the same label as the underlying model. The ranking ability, i.e., the AUC is, on the other hand, based on the calibrated probability estimates.

Table 3: Decision trees - predictive performance

	Accuracy			AUC				Accuracy			AUC		
	NoCal	Platt	VA	NoCal	Platt	VA		NoCal	Platt	VA	NoCal	Platt	VA
balance	.803	.793	.793	.836	.783	.780	user	.871	.871	.871	.783	.664	.666
cars	.951	.945	.945	.932	.918	.922	vehicle	.686	.675	.675	.661	.653	.652
cmc	.521	.492	.492	.639	.643	.666	vowel	.755	.719	.719	.730	.680	.683
cool	.938	.935	.935	.832	.861	.947	wave	.752	.744	.744	.625	.603	.607
ecoli	.828	.812	.812	.699	.750	.763	whole	.582	.550	.550	.661	.648	.652
glass	.689	.680	.680	.682	.647	.653	wine	.884	.908	.908	.701	.729	.775
heat	.979	.978	.978	.901	.864	.937	wineR	.594	.563	.563	.604	.602	.602
image	.948	.945	.945	.838	.803	.918	wineW	.570	.543	.543	.634	.606	.607
iris	.947	.940	.940	.849	.860	.650	yeast	.534	.539	.539	.661	.610	.606
steel	.715	.692	.692	.705	.694	.756	**Mean**	**.754**	**.743**	**.743**	**.729**	**.711**	**.713**
tae	.526	.534	.534	.605	.603	.521	**Mean rank**	**1.40**	**2.30**	**2.30**	**1.50**	**2.25**	**2.25**

Before presenting the aggregated results for the calibration, we take a detailed look at a few data sets exhibiting some typical patterns. Starting with Fig. 2, we can see a fairly common picture where the calibration takes a very poorly calibrated model and reduces the

ECE significantly. Here, the decision tree is extremely overconfident, actually returning a large proportion of estimates very close to 1.0. Both Platt scaling and Venn-Abers lower these estimates substantially, which of course is a good sign since the accuracy is just over 0.5. In fact, there are no calibrated estimates higher than 0.8.

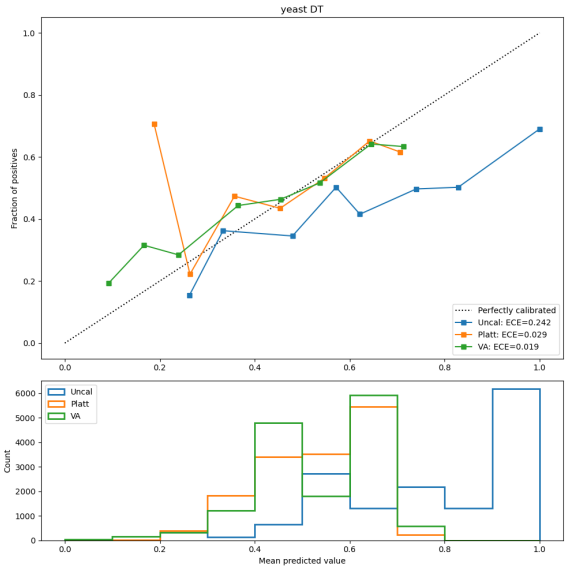

Figure 2: Yeast data set - decision trees

Another similar example, also seen in a number of the data sets, is Fig. 3, where the poorly calibrated tree is significantly improved, but with relatively high ECE:s as the end result. Here it is interesting to see that while the decision tree have many estimates close to 1.0, there are no calibrated estimates from Platt scaling or Venn-Abers higher than 0.9.

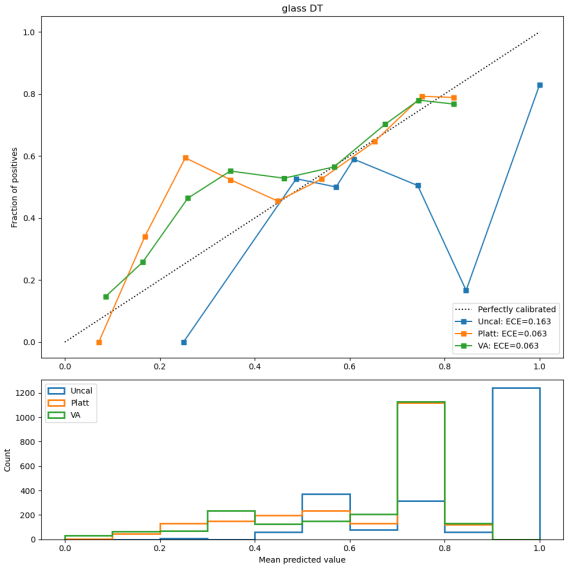

Figure 3: Glass data set - decision trees

The last example for decision trees is Fig. 4 where the already rather well-calibrated tree models are slightly improved using the calibration techniques. With an accuracy over 0.94, it is no surprise that even the calibrated models produce most estimates close to 1.0. Still, both Platt scaling and Venn-Abers lower the extreme estimates from the tree, resulting in better calibration.

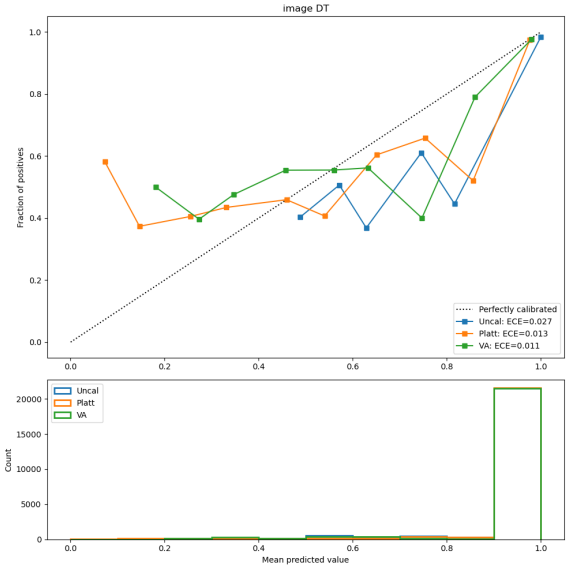

Figure 4: Image data set - decision trees

When looking at the aggregated results in Table 4, there is a clear ordering between the three alternatives. The uncalibrated decision trees often exhibit very high log losses and ECE:s, with the obvious interpretation that they normally are too overconfident in a large majority of all predictions. Platt scaling significantly reduces this, generally resulting in well-calibrated PET:s. While the differences are small in absolute numbers, Venn-Abers actually improves on the Platt scaling results on a large majority of the data sets.

Table 4: Decision trees - calibration

	Log loss			ECE				Log loss			ECE		
	NoCal	Platt	VA	NoCal	Platt	VA		NoCal	Platt	VA	NoCal	Platt	VA
balance	1.536	.402	.391	.072	.070	.044	user	1.695	.345	.345	.077	.024	.030
cars	.242	.124	.111	.021	.017	.014	vehicle	5.455	.591	.593	.219	.035	.025
cmc	3.508	.661	.660	.217	.033	.022	vowel	3.406	.549	.548	.127	.050	.051
cool	.634	.172	.144	.036	.048	.027	wave	5.497	.543	.542	.185	.014	.008
ecoli	2.486	.427	.421	.116	.046	.038	whole	5.839	.652	.648	.247	.024	.031
glass	3.756	.608	.597	.163	.063	.063	wine	2.181	.254	.247	.082	.007	.024
heat	.159	.068	.064	.012	.015	.009	wineR	6.446	.672	.670	.242	.032	.034
image	.579	.162	.154	.027	.013	.011	wineW	6.219	.670	.670	.255	.013	.011
iris	.527	.154	.164	.031	.020	.029	yeast	4.910	.673	.674	.242	.029	.019
steel	4.400	.554	.552	.175	.020	.017	**Mean**	**3.196**	**.448**	**.444**	**.140**	**.031**	**.027**
tae	4.452	.682	.688	.245	.048	.034	**Mean rank**	**3.00**	**1.75**	**1.25**	**2.90**	**1.80**	**1.30**

Summarizing the results for decision trees, the main conclusion is that while the underlying tree models are often extremely poorly calibrated, specifically overconfident, the calibration techniques are able to turn these models into good or very good probabilistic predictors. Regarding the predictive performance, the improved calibration, at least for these rather small data sets, come at a price of lower accuracy and AUC, due to the fact that less labelled data is available for generating the trees.

4.3. Results for random forests

Looking at the predictive performance in Table 5, it can be seen that access to more training data is beneficial also for the random forests. While the differences in absolute numbers are often small on individual data sets, the mean ranks show that the uncalibrated models are significantly more accurate and have a significantly higher AUC than both Platt scaling and Venn-Abers. Comparing Platt scaling to Venn-Abers, Platt scaling has a higher AUC when considering the mean ranks, but it should still be noted that on a few data sets, like Heat and Wine, the AUC is substantially lower than for Venn-Abers.

Table 5: Random forests - predictive performance

	Accuracy			AUC				Accuracy			AUC		
	NoCal	Platt	VA	NoCal	Platt	VA		NoCal	Platt	VA	NoCal	Platt	VA
balance	.832	.851	.851	.947	.909	.906	user	.908	.904	.904	.844	.822	.811
cars	.986	.975	.975	.978	.960	.956	vehicle	.750	.736	.736	.849	.830	.821
cmc	.521	.516	.516	.636	.630	.626	vowel	.972	.933	.933	.924	.896	.887
cool	.950	.945	.945	.938	.891	.921	wave	.854	.852	.852	.818	.815	.812
ecoli	.873	.843	.843	.772	.810	.801	whole	.706	.701	.701	.516	.509	.532
glass	.786	.756	.756	.775	.749	.741	wine	.982	.978	.978	.960	.575	.911
heat	.986	.987	.987	.943	.817	.905	wineR	.701	.661	.661	.750	.713	.709
image	.981	.975	.975	.980	.969	.968	wineW	.704	.660	.660	.770	.728	.725
iris	.953	.945	.945	.911	.914	.931	yeast	.620	.603	.603	.693	.684	.677
steel	.783	.772	.772	.822	.804	.799	**Mean**	**.827**	**.809**	**.809**	**.822**	**.783**	**.805**
tae	.692	.583	.583	.619	.638	.654	**Mean rank**	**1.20**	**2.40**	**2.40**	**1.35**	**2.15**	**2.50**

For some data sets, like Vowel in Fig. 5, the uncalibrated random forest is extremely underconfident. Luckily, calibration using either Platt scaling or Venn-Abers is able to reduce the ECE substantially.

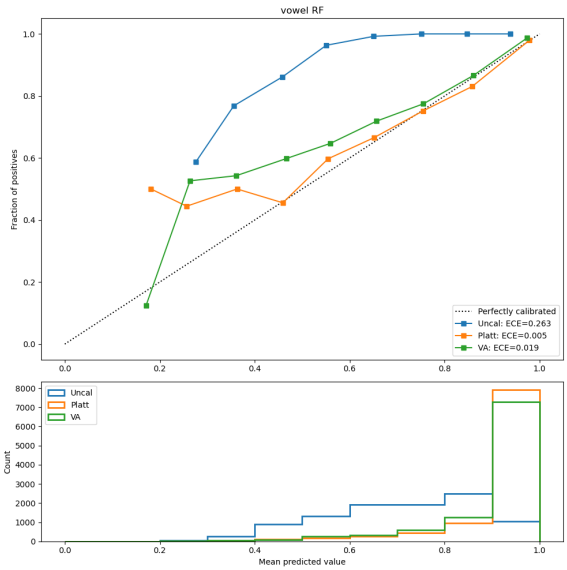

Figure 5: Vowel data set - random forests

Another similar example is the Wave data set, where the random forest is again clearly underconfident, see Fig. 6. Here, both calibration techniques create a number of very confident predictions, leading to almost perfect calibration.

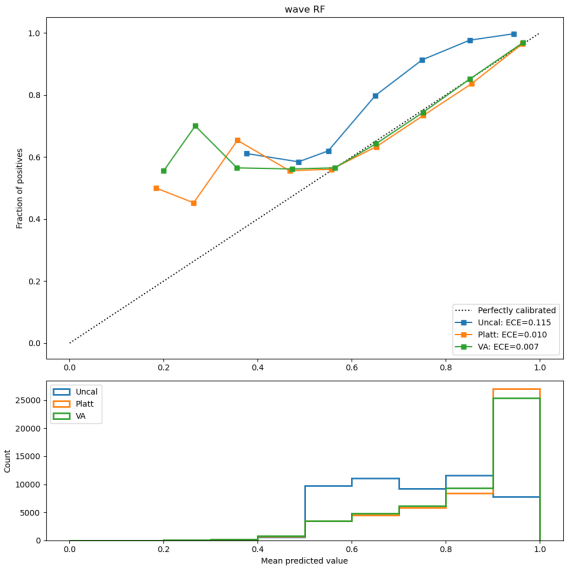

Figure 6: Wave data set - random forests

Actually, the random forests are underconfident on almost all data sets, especially for the higher estimates. The one very different example is the CMC data set, see Fig. 7, where the random forest is actually very overconfident. It is, of course, reassuring to see that both Platt scaling and Venn-Abers are able to calibrate these models too, producing generally lower estimates, thus resulting in significantly lower ECE:s.

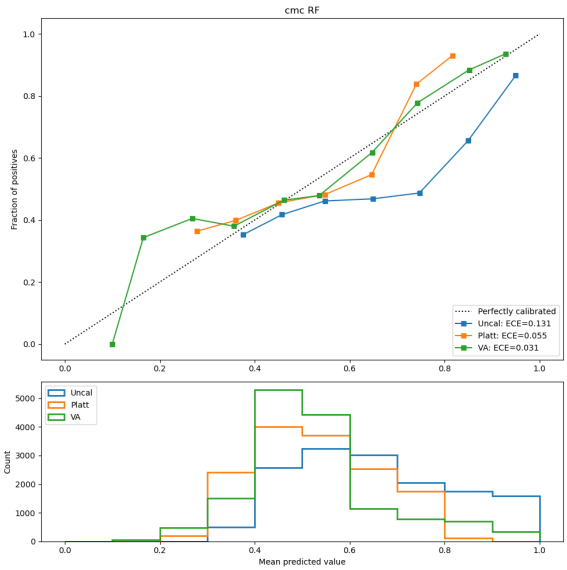

Figure 7: CMC data set - random forests

One of the few data sets where the calibrated models are worse than the uncalibrated is TAE, see Fig. 8. Interestingly enough, while both Platt scaling and Venn-Abers lower the confidences, they are still too confident for the predictions with the highest probability estimates. All-in-all though, ECE:s of $5-8\%$ should probably be considered acceptable on a data set where the model accuracy is under 0.7.

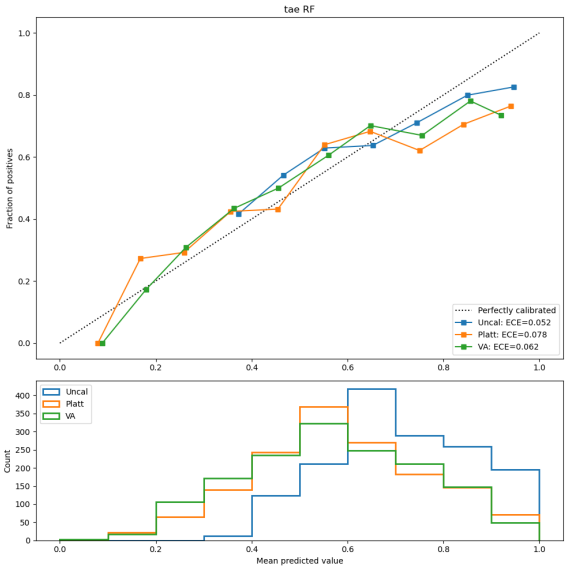

Figure 8: TAE data set - random forests

Considering the aggregated results for random forests in Table 6, we see that the two calibration techniques are able to substantially lower the log loss and significantly improve the

ECE. Looking at both average results and mean ranks, Platt scaling actually outperforms Venn-Abers, even if the differences are small.

Table 6: Random forests - calibration

	Log loss			ECE				Log loss			ECE		
	NoCal	Platt	VA	NoCal	Platt	VA		NoCal	Platt	VA	NoCal	Platt	VA
balance	.240	.264	.264	.075	.051	.047	user	.290	.258	.275	.076	.018	.046
cars	.103	.067	.072	.071	.005	.015	vehicle	.414	.437	.430	.057	.063	.029
cmc	.710	.663	.655	.131	.055	.031	vowel	.367	.172	.180	.263	.005	.019
cool	.154	.138	.133	.005	.015	.019	wave	.387	.334	.336	.115	.010	.007
ecoli	.392	.360	.369	.061	.037	.046	whole	.639	.612	.610	.086	.026	.033
glass	.455	.486	.504	.081	.038	.069	wine	.122	.101	.102	.080	.003	.041
heat	.052	.052	.054	.023	.004	.018	wineR	.528	.575	.576	.037	.032	.016
image	.074	.062	.064	.040	.005	.012	wineW	.516	.560	.555	.056	.041	.014
iris	.166	.143	.145	.017	.014	.048	yeast	.608	.620	.625	.021	.014	.018
steel	.447	.426	.428	.074	.024	.016	**Mean**	**.364**	**.350**	**.352**	**.071**	**.027**	**.030**
tae	.617	.671	.655	.052	.078	.062	**Mean rank**	**2.25**	**1.70**	**2.05**	**2.70**	**1.55**	**1.75**

Summarising the random forest experiment, the main result is that the two calibration techniques are able to successfully calibrate the most often underconfident random forests. Again, the price paid is a small loss in predictive performance.

4.4. Results for xGB

As seen in Table 7, the predictive results are very similar also for xGB. Again, the need for a separate calibration set leads to lower accuracy and AUC. It may be noted that when using xGB as underlying models, Venn-Abers predictors clearly outperform Platt scaling with the regard to AUC.

Table 7: xGB - predictive performance

	Accuracy			AUC				Accuracy			AUC		
	NoCal	Platt	VA	NoCal	Platt	VA		NoCal	Platt	VA	NoCal	Platt	VA
balance	.872	.875	.875	.960	.923	.907	user	.918	.901	.901	.856	.799	.835
cars	.995	.986	.986	.993	.962	.980	vehicle	.762	.745	.745	.842	.762	.792
cmc	.512	.496	.496	.667	.659	.670	vowel	.927	.873	.873	.882	.846	.861
cool	.953	.943	.943	.904	.875	.909	wave	.857	.852	.852	.830	.813	.832
ecoli	.851	.833	.833	.806	.796	.814	whole	.675	.661	.661	.572	.588	.575
glass	.790	.734	.734	.756	.727	.749	wine	.966	.978	.978	.970	.505	.869
heat	.991	.987	.987	.893	.844	.862	wineR	.697	.641	.641	.720	.659	.671
image	.983	.976	.976	.970	.930	.974	wineW	.687	.649	.649	.710	.691	.689
iris	.953	.940	.940	.735	.683	.848	yeast	.594	.580	.580	.681	.662	.656
steel	.804	.788	.788	.835	.813	.811	**Mean**	**.822**	**.800**	**.800**	**.810**	**.760**	**.798**
tae	.656	.563	.563	.618	.658	.659	**Mean rank**	**1.20**	**2.40**	**2.40**	**1.50**	**2.65**	**1.85**

Looking at a few general patterns, Fig. 9 is one common example where the xGB is very overconfident, specifically returning a high number of estimates close to 1.0. Here, both Platt scaling and Venn-Abers move the bulk of estimates to substantially lower estimates, which make sense because the overall accuracy is around 0.7

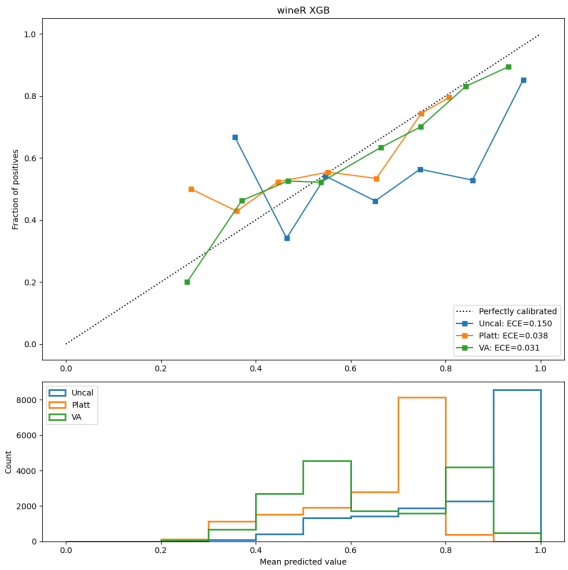

Figure 9: WineR data set - xGB

Unfortunately, and somewhat unexpected, there are a few data sets where the calibration is clearly detrimental. One such case is shown in Fig. 10. In this example, a large majority of all estimates are very high. Venn-Abers and, to a smaller degree, Platt scaling are slightly underconfident on these. Venn-Abers also returns more low estimates than the other techniques.

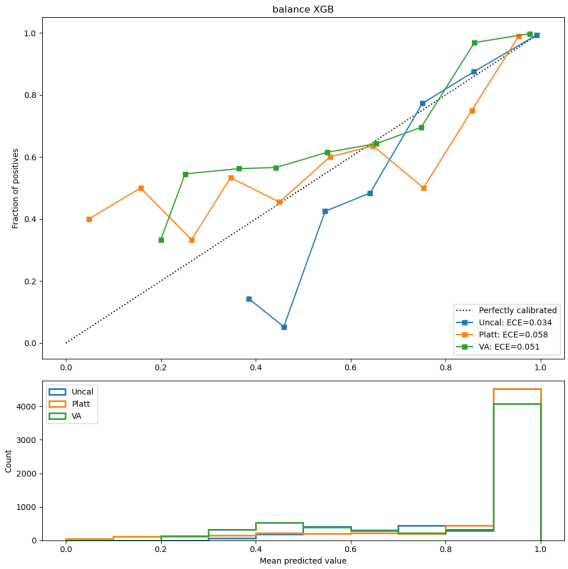

Figure 10: Balance data set - xGB

Looking finally at one of the easiest data sets, we see in Fig. 11 that Venn-Abers again is somewhat underconfident for the estimates close to 1.0. Here, where the accuracy of the model is almost 0.98, Platt scaling actually finds the perfect balance when returning almost

all estimates close to but not equal to 1.0. Venn-Abers, on the other hand, becomes too conservative and is again underconfident for the highest estimates.

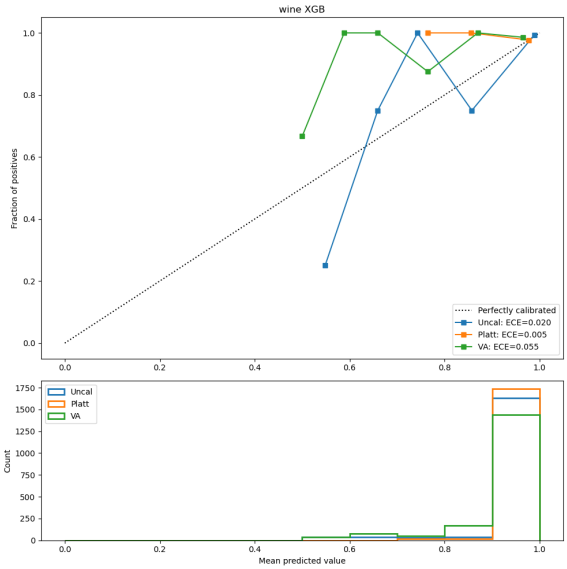

Figure 11: Wine data set - xGB

When analyzing the overall calibration results for xGB in Table 8 we see that for a large majority of the data sets, the calibration does indeed reduce the ECE:s. On average, the ECE:s go from 0.084 to 0.038, which is a substantial difference. Regarding log losses, however, there is only a small gain using calibration for the xGB models. In an outright comparison, Platt scaling lowers the log loss on nine of twenty data sets, compared to the uncalibrated model. The corresponding number for Venn-Abers is eleven.

Table 8: xGB - calibration

	Log loss			ECE				Log loss			ECE		
	NoCal	Platt	VA	NoCal	Platt	VA		NoCal	Platt	VA	NoCal	Platt	VA
balance	.181	.253	.242	.034	.058	.051	user	.249	.291	.264	.041	.018	.051
cars	.020	.044	.041	.011	.005	.011	vehicle	.493	.506	.452	.127	.062	.034
cmc	.780	.660	.638	.195	.069	.047	vowel	.189	.289	.282	.021	.032	.030
cool	.125	.161	.140	.017	.029	.030	wave	.382	.355	.324	.068	.054	.010
ecoli	.429	.381	.370	.089	.055	.058	whole	.966	.629	.641	.199	.019	.048
glass	.532	.525	.519	.118	.076	.059	wine	.075	.108	.120	.020	.005	.055
heat	.036	.058	.060	.003	.010	.024	wineR	.667	.627	.609	.150	.038	.031
image	.037	.076	.059	.004	.006	.013	wineW	.578	.600	.596	.070	.040	.014
iris	.186	.181	.171	.038	.011	.057	yeast	.801	.647	.648	.187	.041	.038
steel	.431	.432	.406	.089	.050	.020	**Mean**	**.401**	**.375**	**.362**	**.084**	**.038**	**.038**
tae	.857	.678	.664	.204	.090	.085	**Mean rank**	**2.00**	**2.35**	**1.65**	**2.35**	**1.80**	**1.85**

Summarizing the xGB experiment, we see that the calibration is again most often able to improve the probability estimates. Specifically, on an aggregated level and using ECE as the main metric, the often seriously overconfident xGB-models are by Platt scaling and

Venn-Abers turned into well-calibrated probabilistic classifiers. Similar to decision trees and random forests, xGB also suffers from having to use smaller training sets, resulting in slightly worse predictive performance.

4.5. Prediction intervals for VAP

One advantage for Venn-Abers compared to other calibration techniques is the potentially more informative probability intervals. Table 9 shows the mean values for the low and high ends of the prediction intervals, together with the empirical accuracies. Starting with the decision trees, we see that for eight of twenty data sets, the empirical accuracy is actually outside the interval. While this is slightly discouraging, it should be remembered that for decision trees, there will be a number of identical estimates since every leaf will have one prediction interval, i.e., each instance falling into that leaf will get the same prediction interval, resulting in a set of very tight intervals. For the random forests and for xGB, however, where the estimates are more granular, we see larger intervals, which, on the other hand, almost always cover the empirical accuracies.

Table 9: VAP probability intervals

	Decision tree			Random forest			xGB		
	Low	High	Acc	Low	High	Acc	Low	High	Acc
balance	.766	.793	.793	.788	.838	.851	.811	.866	**.875**
cars	.938	.947	.945	.956	.975	.975	.973	.991	.986
cmc	.498	.512	**.492**	.511	.539	.516	.512	.545	**.496**
cool	.914	.929	**.935**	.916	.950	.945	.902	.949	.943
ecoli	.782	.818	.812	.796	.870	.843	.760	.847	.833
glass	.591	.654	**.680**	.648	.758	.756	.630	.754	.734
heat	.970	.981	.978	.963	.991	.987	.958	.990	.987
image	.946	.952	**.945**	.959	.976	.975	.964	.981	.976
iris	.900	.958	.940	.872	.982	.945	.876	.977	.940
steel	.704	.713	**.692**	.758	.781	.772	.775	.800	.788
tae	.487	.554	.534	.515	.643	.583	.572	.671	**.563**
user	.849	.880	.871	.839	.905	.904	.851	.914	.901
vehicle	.674	.692	.675	.722	.766	.736	.719	.769	.745
vowel	.670	.686	**.719**	.907	.939	.933	.835	.877	.873
wave	.736	.739	**.744**	.847	.858	.852	.839	.853	.852
whole	.531	.561	.550	.689	.732	.701	.610	.673	.661
wine	.885	.922	.908	.924	.996	.978	.900	.994	.978
wineR	.563	.572	.563	.659	.685	.661	.632	.659	.641
wineW	.530	.534	**.543**	.648	.659	**.660**	.636	.649	.649
yeast	.527	.539	.539	.585	.613	.603	.559	.592	.580
Mean	**.723**	**.747**	**.743**	**.775**	**.823**	**.809**	**.766**	**.818**	**.800**

5. Concluding remarks

We have proposed a novel approach for calibrating multi-class models. Rather than providing probability estimates for all possible labels for each prediction, we here consider the task of estimating the probability that the class label predicted by the underlying model is correct. This avoids a translation of the multi-class problem into a set of binary classification tasks, for which multiple (binary) calibrators have to be generated and merged. In addition to the increased computational complexity, traditional methods also makes the interpretation of the final output more difficult. In contrast, the proposed approach requires only one calibration step, with no need for combining the output of multiple calibrators, hence maintaining the interpretability of the underlying models.

We have presented results from an empirical investigation with 20 data sets and three learning algorithms; decision trees, random forests and xGB. Uncalibrated models have been compared to models calibrated with the proposed approach, using either Platt scaling or Venn-Abers for the actual calibration step. The results show that the quality of the output probabilities, as measured by log loss and expected calibration error, improve compared to not using calibration, and most clearly so for individual decision trees. Platt scaling was observed to perform on par with Venn-Abers, except for PETs, when there was a clear difference in favor of the latter technique.

There are several directions for future research. One suggestion concerns investigating additional underlying models and possibly identifying model classes for which calibration is more effective than for others. Various settings of the hyper-parameters of the techniques, including the fraction of training instances to be used for calibration, could be explored and also the option to use out-of-bag calibration for random forests, like for any other technique that employs bagging. In addition, the usefulness of complementing an underlying predictive model with (well-calibrated) probability estimates of the correctness, as provided by the proposed approach, remains to be evaluated. The employed performance metrics give an indication of how accurate the probability estimates are, but does not demonstrate the actual gains of the approach, e.g., in terms of increased utility or reduced costs.

Acknowledgements

UJ and TL were partly funded by the Swedish Knowledge Foundation (DATAKIND 20190194). HB was partly funded by the Swedish Foundation for Strategic Research (CDA, grant no. BD15-0006).

References

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Alexander Gammerman, Volodya Vovk, and Vladimir Vapnik. Learning by transduction. In *Uncertainty in artificial intelligence*, pages 148–155. Morgan Kaufmann Publishers Inc., 1998.

Ulf Johansson, Tuwe Löfström, and Henrik Boström. Calibrating probability estimation trees using venn-abers predictors. In *Proceedings of the 2019 SIAM International Conference on Data Mining, SDM 2019, Calgary, Alberta, Canada, May 2-4, 2019.*, pages 28–36, 2019.

Antonis Lambrou, Ilia Nouretdinov, and Harris Papadopoulos. Inductive venn prediction. *Annals of Mathematics and Artificial Intelligence*, 74(1):181–201, 2015.

Valery Manokhin. Multi-class probabilistic classification using inductive and cross Venn–Abers predictors. In *Proceedings of the Sixth Workshop on Conformal and Probabilistic Prediction and Applications*, volume 60 of *Proceedings of Machine Learning Research*, pages 228–240, Stockholm, Sweden, 2017. PMLR.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

Foster Provost and Pedro Domingos. Tree induction for probability-based ranking. *Mach. Learn.*, 52(3):199–215, 2003.

Vladimir Vovk and Ivan Petej. Venn-abers predictors. *arXiv preprint arXiv:1211.0025*, 2012.

Vladimir Vovk, Glenn Shafer, and Ilia Nouretdinov. Self-calibrating probability forecasting. In *Advances in Neural Information Processing Systems*, pages 1133–1140, 2004.

Jonathan Wenger, Hedvig Kjellström, and Rudolph Triebel. Non-parametric calibration for classification. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 178–190. PMLR, 26–28 Aug 2020.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Proc. 18th International Conference on Machine Learning*, pages 609–616, 2001.

Bianca Zadrozny and Charles Elkan. 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, page 694–699, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 158113567X. doi: 10.1145/775047.775151.