Calibrating Probability Estimation Trees using Venn-Abers Predictors*

Ulf Johansson† Tuwe Löfström† Henrik Boström‡

Abstract

Class labels output by standard decision trees are not very useful for making informed decisions, e.g., when comparing the expected utility of various alternatives. In contrast, probability estimation trees (PETs) output class probability distributions rather than single class labels. It is well known that estimating class probabilities in PETs by relative frequencies often lead to extreme probability estimates, and a number of approaches to provide more well-calibrated estimates have been proposed. In this study, a recent model-agnostic calibration approach, called Venn-Abers predictors is, for the first time, considered in the context of decision trees. Results from a large-scale empirical investigation are presented, comparing the novel approach to previous calibration techniques with respect to several different performance metrics, targeting both predictive performance and reliability of the estimates. All approaches are considered both with and without Laplace correction. The results show that using Venn-Abers predictors for calibration is a highly competitive approach, significantly outperforming Platt scaling, Isotonic regression and no calibration, with respect to almost all performance metrics used, independently of whether Laplace correction is applied or not. The only exception is AUC, where using non-calibrated PETs together with Laplace correction, actually is the best option, which can be explained by the fact that AUC is not affected by the absolute, but only relative, values of the probability estimates.

1 Introduction

After more than thirty years, decision trees are still used in many real-world data analytics projects. Technically, decision trees are quite accurate, fast to train and require a minimum of parameter tuning. They are also readily available in all major data mining tools and languages. The main reason for employing decision trees is, however, that the models are *interpretable*, i.e., it is possible to inspect the model in order to gain insights, and to understand the logic behind individual predictions.

Today, when predictive modeling is increasingly used for not only decision support, but also automated decision making, *trust* in the generated models becomes vital. Obviously, interpretability is often associated with user acceptance and trust, see e.g., [5]. In fact, in several domains with special requirements on the documentation of decisions made for legal or safety purposes, interpretable models are even mandatory [1]. We believe that these situations will become ever more frequent when AI and automated agents, in a large variety of domains, will be required to make more and more decisions based on predictive models. Consequently, interpretability is, and probably will be for a foreseeable future, a prioritized area for machine learning and data mining research, see e.g., [6].

However, interpretability may not be a sufficient condition for enabling trust in predictive models, as users must also be able to have *confidence* in the predictions from the model. Traditionally, confidence has been more or less synonymous with that the model should be as accurate as possible, i.e., that the predictions are correct. Accuracy, and similar metrics are, however, rather blunt, specifically as they only assess how well the model performs on average, when making a number of predictions. We argue that an accurate model with the ability to distinguish between predictions where it is certain and not leads to increased trust. Specifically, we suggest the use of *probabilistic predictors*, outputting not only the predicted class label, but also a probability distribution over the possible classes.

Probabilistic prediction requires *well-calibrated* models, i.e., that predicted class probabilities reflect the true, underlying probabilities. If this is not the case, the probabilistic predictor instead becomes misleading. On the other hand, an automated decision-maker, utilizing a well-calibrated probabilistic predictor, is a very strong option. Specifically, if combined with utilities, it will produce optimal decisions, according to Bayesian decision theory.

To be truly useful, however, probabilistic predictors must be not only well-calibrated, i.e., *valid*, but also *specific*. Consider a probabilistic predictor simply using the relative frequencies of the class labels in a training set as the prediction for all test instances. That

*This work was supported by the Swedish Knowledge Foundation through the project Data Analytics for Research and Development (20150185).

†Dept. of Computer Science and Informatics, Jönköping University, Sweden. {ulf.johansson, tuwe.lofstrom}@ju.se

‡School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden, bostromh@kth.se

predictor will be (asymptotically) perfectly calibrated (unbiased) as long as the data sets are i.i.d. Obviously, such a model, which in fact ignores properties (features) of the instance to be predicted, is not very informative, see [19]. Instead, the predictions should be as relevant as possible to the specific test instance, i.e., we want *predictive efficiency*, with maintained validity.

Decision trees are inherently capable of producing probabilistic predictions. Such decision trees are called *Probability Estimation Trees* (PETs) [15]. In the most basic approach, the relative frequencies of the different labels from the training instances reaching a specific leaf are used as the probabilistic prediction for all test instances falling in that leaf. In [15], Provost and Domingos suggested to instead use Laplace estimates in order to produce what they called *well-behaved PETs*. While the Laplace correction will improve the probability estimates, they will still typically overestimate the true probabilities to a large extent, see e.g., [10].

There also exist several model-agnostic methods for calibrating probabilistic predictions; the two most well-known are *Platt scaling* [14] and *isotonic regression* [21]. Both techniques have been successfully applied to many different predictive models, including naïve Bayes, support-vector machines and boosted decision trees [13]. For decision trees, however, these calibration techniques have often been ineffective, see [13].

Recently, so called *Venn predictors* [20], have been suggested as an alternative approach to calibrating probabilities from decision trees [10]. Venn predictors are, under the standard i.i.d. assumption, perfectly calibrated (unbiased) [20]. Venn predictors are *multiprobabilistic* predictors, i.e., they output a set of probabilistic predictions, with one of them being the valid one, instead of a single probabilistic prediction. Somewhat simplified, these multiprobabilistic predictions can be regarded as probability intervals for each label. Consequently, the size of these intervals reflects the confidence in the estimate.

In [10], Johansson et al. compared Venn predictors to Platt scaling and isotonic regression for calibrating PETs on two-class problems. The empirical investigation conclusively showed that the probability estimates from the Venn predictor were the most exact. However, the predictions produced by the setup used for the Venn predictors were not very specific, in fact only two (one for each class) distinct probability intervals were used in each tree. With this in mind, the overall purpose of this paper is to suggest and evaluate so-called *Venn-Abers predictors* [19] for producing more specific, but still valid, predictions from PETs. In addition, we will make a comprehensive investigation of whether it is beneficial to perform the calibration (with Venn-Abers, Platt scaling or isotonic regression) using the Laplace estimates instead of the raw relative frequencies.

In the next section, we first introduce probabilistic prediction and describe probability estimation trees, before presenting the considered calibration techniques. In Section 4, we outline the experimental setup, which is followed by the experimental results in Section 5. Finally, we summarize the main conclusions and suggest some directions for future work in Section 6.

2 Background

2.1 Probabilistic prediction A probabilistic predictor outputs a probability distribution of the label, given the training set and the test instance. The goal is to obtain *validity*, i.e., that probability distributions from the predictor perform well against statistical tests based on subsequent observation of the labels. In particular, we are interested in *calibration*:

$$p(c_j \mid p^{c_j}) = p^{c_j}, \tag{2.1}$$

where p^{c_j} is the probability estimate for class j. An important impossibility result is that validity cannot be achieved for probabilistic prediction in a general sense, see e.g., [8].

2.2 Probability Estimation Trees The two most famous decision tree algorithms are C4.5/C5.0 [16] and CART [4]. While they differ in the induction and the split criterion, the representational language, at least if restricted to binary splits, is identical. With this in mind, all calibration techniques applicable to decision trees can be used for both CART and C4.5/C5.0.

As mentioned in the introduction, the most straightforward way to obtain class probabilities from PETs is to use the *relative frequency*; i.e., the proportion of training instances corresponding to a specific class in the leaf where the test instance falls. In equation (2.2) below, the probability estimate $p_i^{c_j}$, based on relative frequencies, is defined as

$$p_i^{c_j} = \frac{g(i,j)}{\sum_{k=1}^{C} g(i,k)} \tag{2.2}$$

where $g(i,j)$ is the number of instances belonging to class j that falls in the same leaf as instance i, and C is the number of classes.

Often, however, the relative frequencies are not used as the probability estimates, but instead a *Laplace correction* is applied:

$$p_i^{c_j} = \frac{1 + g(i,j)}{C + \sum_{k=1}^{C} g(i,k)} \tag{2.3}$$

The main reason for this smoothening is that the relative frequency estimate does not consider the number of training instances reaching a specific leaf. Intuitively, a leaf containing many training instances is a more robust estimator of class membership probabilities.

In addition to using Laplace estimates, Provost and Domingos in [15], also changed the C4.5 algorithm by turning off both pruning and the collapsing mechanism, which obviously led to substantially larger trees. These modifications produced much better PETs; for more details see the original paper.

2.3 Platt scaling The calibration technique Platt scaling [14] fits a sigmoid function to the training set or, most often, a specific calibration set. The function is

$$\hat{p}(c \mid s) = \frac{1}{1 + e^{As+B}}, \tag{2.4}$$

where $\hat{p}(c \mid s)$ gives the probability that an example belongs to class c, given that it has obtained the score s. A and B are found by a gradient descent search, minimizing a particular loss function suggested in [14].

Platt recommends that the parameters of the sigmoid function are optimized on a specific calibration set, not used for generating the model. In addition, regularization is most often applied by using the following target values (where k_+ and k_- are the number of calibration instances labeled 1 and 0, respectively) instead of 0 and 1:

$$t_+ := \frac{k_+ + 1}{k_+ + 2} \tag{2.5}$$

$$t_- := \frac{1}{k_- + 2} \tag{2.6}$$

Using these values, Platt estimates will never be exactly 0 or 1, so there is no risk for infinite log losses. It should be noted that while Platt scaling often works well in practice, it implicitly assumes that the optimal calibration curve is indeed a sigmoid, which of course may not be true.

2.4 Isotonic regression Zadrozny and Elkan in [21], suggested isotonic regression for calibration. Here, the calibration function, which is assumed to be *isotonic*, i.e., non-decreasing, is a step-wise regression function, that can be learned by an algorithm known as the *pair-adjacent violators algorithm*. Starting with a set of input probability intervals, whose borders are the scores in the training set, it repeatedly merges adjacent intervals for which the lower interval contains a higher (or equally high) fraction of examples belonging to the positive class. When no such pair of intervals can be found, the algorithm terminates and outputs a function that for each input probability interval returns the fraction of positive examples in the calibration set in that interval. In contrast to Platt scaling, there is normally no regularization present in isotonic regression, so infinite log losses (which of course happens if the estimate is 0 and the true label is 1, or vice versa) must be expected. While there are some tricks that can be used to avoid this problem, like adding one dummy instance with estimate $+\infty$ and label 0 and another dummy instance with estimate $-\infty$ and label 1, there is no standard procedure for regularization in isotonic regression.

3 Venn predictors

Venn predictors [20] circumvent the general impossibility result regarding validity for probabilistic prediction in two ways: (i) multiple probabilities for each label are output, with one of them being the valid one; (ii) statistical tests for validity are restricted to calibration. In practice, the probabilities should be matched by observed accuracy. As an example, if we make a number of predictions with the probability estimate 0.9, we expect these predictions to be correct in about 90% of the cases.

Venn predictors are used on top of standard predictive models, referred to as the *underlying models*. The key idea of Venn prediction is to somehow use the underlying model to divide all calibration examples into a number of *categories*. When doing predictions, the underlying model is again used, in an identical way as for the calibration instances, to determine the category for each test instance. Finally, the estimated class probabilities for a test instance is calculated from the relative label frequencies of the calibration instances belonging to that category. To obtain validity, this calculation must include the test instance to be predicted. However, since the true label is, per definition, not known for the test instance, all possible labels must be considered, which results in a set of C label probability distributions, where C is the number of possible labels.

Venn predictors were introduced in a *transductive* setting, where one underlying model has to be trained for every possible label of each test instance. This is, of course, computationally inefficient, but it also means that there is no fixed model that can be inspected and analyzed, thus severely hampering interpretability. In fact, even when predicting just one test instance, there would be C models involved, where C is the number

of possible labels. There exists, however, an *inductive* version of Venn prediction [11], that uses only one, fixed, underlying model. We now describe inductive Venn predictors, and the concept of multiprobability prediction, following the presentation in [9].

To generate an inductive Venn predictor, the labeled examples are split into two parts, the *proper training set*, used to train the underlying model, and the *calibration set* used to estimate label probabilities for each new test example.

Assume we have a training set of the form $\{z_1, \ldots, z_l\}$ where each instance $z_i = (x_i, y_i)$ consists of two parts; an *object* x_i and a *label* y_i. Let the proper training set be $\{z_1, \ldots, z_q\}$ and the calibration set $\{z_{q+1}, \ldots, z_l\}$. When given a new test object x_{l+1}, Venn prediction estimates the probability that $y_{l+1} = Y_j$, for each Y_j in the set of possible labels $Y_j \in \{Y_1, \ldots, Y_c\}$. In the inductive version, the calibration instances are divided into a number of *categories*, based on a so-called *Venn taxonomy*, and then the relative frequency of label $Y_j \in \{Y_1, \ldots, Y_c\}$ in each category is used to estimate label probabilities for test instances falling into that category. Most Venn taxonomies are based on the underlying model, where different Venn taxonomies lead to different Venn predictors. Typically, the output of the underlying model is somehow used to assign each calibration instance (x_i, y_i) and test object x_i, into one of the categories. The most basic Venn taxonomy, used in for instance [10], simply puts all examples predicted with the same label into the same category.

When estimating label probabilities for a test instance, the category of that instance is first determined using the underlying model, in an identical way as for the calibration instances. Then, the label frequencies of the calibration instances in that category are used to calculate the label estimates. To obtain validity, the test instance z_{l+1} is included in this calculation. However, since the true label y_{l+1} is not known for the test object x_{l+1}, all possible labels $Y_j \in \{Y_1, \ldots, Y_c\}$ are used to create a set of label probability distributions, which is the output of the Venn predictor. Most often, these distributions are replaced by a compact representation consisting of the lower $L(Y_j)$ and upper $U(Y_j)$ probability estimates for each label Y_j.

The lower and upper probability estimates are defined by:

$$L(Y_j) = \frac{|\{(x_m, y_m) \in Z_k \mid y_m = Y_j\}|}{|Z_k| + 1} \tag{3.7}$$

and:

$$U(Y_j) = \frac{|\{(x_m, y_m) \in Z_k \mid y_m = Y_j\}| + 1}{|Z_k| + 1} \tag{3.8}$$

where k is the category assigned to the test object x_{l+1} by the Venn taxonomy, and Z_k is the set of calibration instances belonging to category k.

A prediction $\hat{y}_{l+1}$ for x_{l+1} is made using the lower and upper probability estimates:

$$\hat{y}_{l+1} = \max_{Y_j \in \{Y_1, \ldots, Y_c\}} L(Y_j) \tag{3.9}$$

The final output of a Venn predictor is the above prediction $\hat{y}_{l+1}$ together with the probability interval:

$$[L(\hat{y}_{l+1}), U(\hat{y}_{l+1})] \tag{3.10}$$

The multiprobability predictions produced by Venn predictors are automatically valid, regardless of the taxonomy used [18]. Still, the taxonomy is important since it will affect both the accuracy of the Venn predictor and the size of the prediction interval. Obviously, probability estimates as close to one or zero as possible are preferred, and smaller probability intervals are considered to be more informative. Furthermore, it should be noted that the more categories that are used in the taxonomy, the more specific the predictions will be. As an example, in a two-class problem, the basic taxonomy that puts all the examples predicted with the same label into the same category will have exactly two categories, so the Venn predictor will for every test object output one of only two possible prediction intervals. On the other hand, if we have too many categories, the calibration will depend on just a few instances, resulting in larger intervals.

3.1 Venn-Abers predictors As described above, the key challenge of Venn predictors is to find a suitable taxonomy. In so-called *Venn-Abers predictors*, this problem is handled by automatic optimization of the taxonomy using isotonic regression [19]. Venn-Abers predictors are Venn predictors, i.e., they inherit the validity guarantee of Venn predictors, but they are in the basic setting only applicapble to two-class problems. Thus, a Venn-Abers predictor retains validity, while providing specific predictions.

Since Venn-Abers predictors are restricted to two-class problems, they can regard the underlying models as *scoring classifiers*, i.e., when an underlying model makes a prediction for a test object, the output is a *prediction score* $s(x)$, where a higher value indicates a larger belief in the label 1. For scoring classifiers applied to two-class problem (using labels 0 and 1) a prediction is obtained by comparing the score to a fixed threshold t, and predicting the label 1 if $s(x) > t$, and 0 otherwise. An alternative to find and use a fixed threshold c is, however, to calibrate an increasing function g using a

number of predictions scores with known true targets. After such a calibration, $g(s(x))$ should be interpreted as the probability that the label for x is 1.

Venn-Abers predictors use isotonic regression, as described in Section 2.4 above, for this calibration. A multiprobabilistic prediction from a Venn-Abers predictor is, in the inductive setting, produced as follows:

Let s_0 be the scoring function for $\{z_{q+1}, \ldots, z_l, (x_{l+1}, 0)\}$, s_1 be the scoring function for $\{z_{q+1}, \ldots, z_l, (x_{l+1}, 1)\}$, g_0 be the isotonic calibrator for $\{(s_0(x_{q+1}), y_{q+1}), \ldots, (s_0(x_l), y_l), (s_0(x_{l+1}), 0)\}$ and g_1 be the isotonic calibrator for $\{(s_1(x_{q+1}), y_{q+1}), \ldots, (s_1(x_l), y_l), (s_1(x_{l+1}), 1)\}$. Then the probability interval for $y_{l+1} = 1$ is $[g_0(s_0(x_{l+1})), g_1(s_1(x_{l+1}))]$

When a Venn-Abers predictor is applied on top of a decision tree, the number of categories is of course dynamic. But, at the same time, when the inductive variant is used, all instances falling in the same leaf will obtain the same estimate, and these estimates can be determined from the calibration set. So the resulting model is a fixed decision tree, available for inspection and analysis, where each leaf contains a specific prediction, consisting of a label and an associated confidence (a probability interval). Clearly this is a very informative model.

4 Method

In the empirical investigation, we look at different ways of producing probability estimates from standard decision trees. Since all experiments were performed in MatLab, the decision trees were induced using the MatLab version of CART, called *ctree*. Here, all parameter values were left at their default values, leading to fairly large trees, which of course is consistent with the recommendations in [15]. The 25 data sets used are all two-class problems, publicly available from either the UCI repository [2] or the PROMISE Software Engineering Repository [17]. In the experimentation, standard 10x10-fold cross-validation was used, so all results reported are averaged over the 100 folds.

For the actual calibration, we compared using inductive Venn-Abers predictors to Platt scaling and isotonic regression, as well as using no external calibration, i.e., the raw estimates from the tree model. Naturally, all three methods employing calibration require a separate labeled data set (the *calibration set*) not used for learning the trees; here 2/3 of the training instances were used for the tree induction and 1/3 for the calibration. In summary, we compare the following four approaches:

- **Tree**: The probability estimates from the tree. Since this approach does not need any external calibration, all training data was used for generating the tree.
- **Platt**: Standard Platt scaling where the logistic regression model was learned on the calibration set.
- **Iso**: Standard isotonic regression using a calibration set.
- **VAP**: Inductive Venn-Abers predictors.

For all approaches, we also evaluate using either the raw relative frequencies or Laplace estimates.

For the analysis, a number of metrics are used. First we look at both accuracy and area under the ROC curve (AUC) for the resulting PETs. While accuracy is based only on the final classification, AUC measures the ability to rank instances according to how likely they are to belong to a certain class, and can be interpreted as the probability of ranking a true positive instance ahead of a false positive.

The quality of the probability estimates will be evaluated using the log loss function defined as:

$$\lambda_{log} = \begin{cases} -\log p & \text{if } y = 1 \\ -\log(1-p) & \text{if } y = 0 \end{cases} \tag{4.11}$$

where log is the binary logarithm, and p is the probability estimate for the label 1. We will also use the *Brier loss*

$$\lambda_{Br} = (y-p)^2 \tag{4.12}$$

which can be used to compare the quality of the estimates even for instances where the log loss is infinite, i.e., when an estimate is categorical, but wrong. The Brier loss can be decomposed into three terms called *uncertainty*, *resolution* and *reliability*. In practice, this is done by dividing the range of probability values into a number of K intervals and represent each interval $1, 2, ..., K$ by a corresponding typical probability value r_k, see [12]. Specifically, the reliability term, defined in 4.13, measures how close the probability estimates are to the true probabilities, making it another measurement of how well-calibrated the estimates are.

$$Reliability = \frac{1}{N}\sum_{k=1}^{K} n_k(r_k - \phi_k)^2, \tag{4.13}$$

where n_k is the number of instances in interval k, r_k is the mean probability estimate for the positive class over the instances in interval k and ϕ_k is the proportion of instances actually belonging to the positive class

in interval k. In the experimentation, the number of intervals K was set to 100. Here, it must be noted that for reliability, lower values are actually better.

When it is necessary to obtain a single probability estimate from the Venn-Abers probability interval (p_0, p_1), in order to compare it to the other approaches, the suggestion in [19] was used:

$$p = \frac{p_1}{1 - p_0 + p_1} \tag{4.14}$$

thus providing a regularized value where the estimate is moved towards the neutral value 0.5.

5 Results

Starting with the predictive performance of the PETs, Table 1 below shows the obtained accuracies.

Table 1: Accuracy

	RF				LaP			
	Tree	Platt	Iso	VAP	Tree	Platt	Iso	VAP
colic	.787	.799	.819	.818	.787	.800	.835	.834
creditA	.830	.830	.834	.834	.830	.830	.838	.838
diabetes	.710	.709	.712	.712	.710	.713	.718	.719
german	.610	.703	.701	.701	.610	.703	.700	.700
haber	.664	.712	.700	.701	.664	.712	.701	.703
heartC	.733	.754	.759	.761	.733	.755	.763	.764
heartH	.767	.765	.767	.768	.767	.769	.774	.775
heartS	.760	.756	.755	.757	.760	.757	.756	.758
hepati	.769	.791	.782	.784	.769	.791	.784	.788
iono	.879	.877	.880	.880	.879	.878	.883	.883
je4042	.699	.691	.697	.695	.699	.692	.705	.703
je4243	.603	.605	.611	.613	.603	.611	.616	.616
kc1	.684	.734	.730	.730	.684	.736	.737	.737
kc2	.727	.739	.746	.747	.727	.752	.766	.767
kc3	.834	.866	.861	.862	.834	.866	.860	.861
liver	.639	.622	.626	.626	.639	.630	.639	.639
pc1req	.664	.614	.624	.626	.664	.620	.632	.635
pc4	.871	.873	.874	.874	.871	.880	.881	.882
sonar	.715	.704	.706	.705	.715	.706	.706	.707
spect	.850	.887	.888	.884	.850	.888	.888	.884
spectf	.740	.785	.784	.783	.740	.787	.787	.786
transf.	.704	.730	.726	.727	.704	.734	.732	.733
ttt	.927	.912	.918	.918	.927	.912	.918	.919
wbc	.915	.912	.914	.914	.915	.912	.915	.915
vote	.841	.840	.843	.844	.841	.841	.846	.846
Mean	**.757**	**.768**	**.770**	**.770**	**.757**	**.771**	**.775**	**.776**
Mean rank	**2.92**	**2.84**	**2.26**	**1.98**	**3.16**	**2.84**	**2.24**	**1.76**

While there are only small differences in absolute numbers, it is clearly beneficial for all three calibration techniques to use the Laplace estimates. Furthermore, the ordering between the setups is the same for using both the relative frequencies and the Laplace estimates; the VAP is the most accurate followed by Iso, Platt and Tree. To determine any statistically significant differences, we used Friedman tests [7], followed by Bergmann-Hommel's dynamic procedure [3] to establish all pairwise differences. In this analysis, we looked at the results for using the relative frequencies and the Laplace estimates separately. When using the relative frequencies as input to the calibration, there were no statistically significant differences at $\alpha = 0.05$. For the Laplace estimates, however, VAP was found to be significantly more accurate that Platt and Tree, while Iso was significantly more accurate than Tree.

Looking at the ranking ability, we see in Table 2 below, that using Laplace estimates is advantageous for all methods.

Table 2: AUC

	RF				LaP			
	Tree	Platt	Iso	VAP	Tree	Platt	Iso	VAP
colic	.802	.805	.819	.822	.860	.839	.850	.853
creditA	.862	.860	.874	.875	.912	.897	.908	.909
diabetes	.730	.707	.721	.723	.785	.748	.772	.772
german	.542	.515	.542	.542	.561	.529	.558	.558
haber	.585	.545	.571	.574	.608	.564	.591	.593
heartC	.776	.779	.788	.792	.825	.806	.817	.821
heartH	.786	.780	.798	.801	.833	.809	.826	.829
heartS	.786	.776	.782	.786	.830	.801	.808	.812
hepati	.650	.659	.673	.687	.706	.677	.696	.712
iono	.879	.887	.892	.895	.920	.908	.916	.918
je4042	.730	.719	.737	.739	.783	.746	.770	.772
je4243	.628	.619	.632	.633	.659	.637	.657	.657
kc1	.595	.587	.593	.594	.616	.600	.612	.612
kc2	.688	.694	.726	.729	.782	.735	.785	.786
kc3	.680	.609	.647	.655	.731	.639	.698	.705
liver	.636	.617	.622	.623	.669	.632	.641	.642
pc1req	.692	.641	.650	.653	.694	.648	.658	.659
pc4	.755	.776	.814	.816	.894	.847	.891	.893
sonar	.724	.720	.718	.722	.772	.740	.752	.756
spect	.543	.479	.523	.546	.577	.495	.542	.568
spectf	.642	.610	.630	.637	.698	.636	.680	.688
transf.	.646	.624	.645	.646	.686	.653	.676	.677
ttt	.971	.952	.959	.961	.979	.962	.967	.968
wbc	.942	.933	.934	.937	.957	.946	.945	.949
vote	.877	.866	.880	.884	.901	.888	.893	.896
Mean	**.726**	**.710**	**.727**	**.731**	**.770**	**.735**	**.756**	**.760**
Mean rank	**2.24**	**3.72**	**2.56**	**1.48**	**1.12**	**3.96**	**3.00**	**1.92**

Interestingly enough, the highest AUC is obtained without any additional calibration. This may be explained by the fact that more training data is available when not performing calibration, which hence leads to larger trees, which in turn has been shown to be beneficial when targeting AUC [15]. Moreover, it should be noted that AUC is not affected by the absolute, but only relative, values of the probability estimates. Hence, even with poorly calibrated estimates, the AUC may be high, as long as output estimates have a high correlation with the true probabilities. Statistical tests show

that when using relative frequencies, VAP obtained a significantly higher AUC than all other setups, while Tree and Iso significantly outperformed Platt. For the Laplace estimates, all differences are statistically significant at $\alpha = 0.05$.

Turning to the quality of the estimates, Table 3 below shows the difference between the estimates and the empirical accuracy on each data set.

Table 3: Difference between estimated and observed accuracy

	RF				LaP			
	Tree	Platt	Iso	VAP	Tree	Platt	Iso	VAP
colic	.153	-.007	.007	-.011	.114	.006	.009	-.013
creditA	.119	.000	.010	-.004	.084	.013	.016	-.003
diabetes	.219	.006	.018	.009	.164	.016	.022	.007
german	.246	.001	.003	-.002	.181	.001	.005	-.002
haber	.201	.010	.029	.012	.137	.011	.032	.009
heartC	.188	-.007	.017	-.006	.142	.004	.027	-.003
heartH	.149	.000	.027	.005	.107	.007	.032	.003
heartS	.159	-.007	.025	-.001	.116	.003	.035	.003
hepati	.172	.016	.042	.011	.131	.018	.045	.008
iono	.106	-.009	.008	-.011	.072	-.002	.016	-.014
je4042	.203	.001	.029	.009	.148	.014	.036	.011
je4243	.285	.016	.038	.026	.215	.019	.044	.026
kc1	.232	.002	.008	.003	.174	.001	.011	.002
kc2	.221	.017	.028	.012	.169	.013	.027	.003
kc3	.122	.006	.015	-.005	.087	.006	.018	-.009
liver	.248	.013	.036	.024	.186	.012	.036	.019
pc1req	.185	.031	.068	.027	.119	.027	.067	.022
pc4	.101	.009	.009	.003	.074	.009	.009	-.001
sonar	.257	-.007	.019	.000	.202	.000	.040	.011
spect	.078	.002	.006	-.020	.040	.001	.006	-.023
spectf	.218	.014	.020	.005	.171	.013	.024	.002
transf.	.169	.012	.022	.008	.114	.011	.022	.004
ttt	.034	.006	.006	-.007	.003	.008	.007	-.018
wbc	.054	-.007	.008	-.013	.031	-.003	.011	-.015
vote	.075	.009	.019	-.003	.048	.014	.020	-.005
Mean	**.168**	**.005**	**.021**	**.003**	**.121**	**.009**	**.025**	**.001**

Starting with the Tree estimates, we see that they systematically overestimate the true accuracies. In fact, even the Laplace estimate is on average more than twelve percentage points too optimistic, i.e., it is obviously misleading. In this study, all estimates from isotonic regression are larger than the true accuracies. Even if these differences are much smaller than the estimates without calibration, the fact is that Iso too turned out to be intrinsically optimistic, i.e., misleading. For both Platt scaling and the VAP, there is no inherent tendency to overestimate or underestimate the accuracy. In fact, even when looking at each and every data set, the probability estimates are often very close to the true accuracies.

As seen in Table 3 below, isotonic regression and regular PETs without Laplace correction, as expected, suffer from infinite log losses. Most importantly, VAP obtained significantly lower log losses than the other setups, both when using relative frequencies and Laplace estimates. In fact, doing a direct comparison with the second best setup (i.e., Platt) we see that when starting from the Laplace estimates, the VAP achieved a lower log loss on all data sets. Also regarding log loss, all setups benefit from using Laplace estimates instead of relative frequencies for the calibration.

Table 4: Logloss

	RF				LaP			
	Tree	Platt	Iso	VAP	Tree	Platt	Iso	VAP
colic	∞	.706	∞	.679	.716	.666	∞	.621
creditA	∞	.630	∞	.605	.595	.587	∞	.536
diabetes	∞	.838	∞	.826	.922	.814	∞	.775
german	∞	.880	∞	.880	1.064	.879	∞	.876
haber	∞	.862	∞	.862	.973	.857	∞	.856
heartC	∞	.797	∞	.783	.851	.769	∞	.744
heartH	∞	.758	∞	.735	.777	.730	∞	.696
heartS	∞	.797	∞	.788	.807	.771	∞	.755
hepati	∞	.698	∞	.685	.820	.688	∞	.673
iono	∞	.518	∞	.502	.521	.487	∞	.455
je4042	∞	.889	∞	.872	.923	.863	∞	.825
je4243	∞	.964	∞	.960	1.158	.954	∞	.942
kc1	∞	.817	.818	.817	1.042	.813	∞	.811
kc2	∞	.766	∞	.744	.883	.744	∞	.685
kc3	∞	.546	∞	.532	.618	.538	∞	.518
liver	∞	.954	∞	.955	1.097	.945	∞	.941
pc1req	∞	.973	∞	.962	.989	.967	∞	.957
pc4	∞	.457	.430	.428	.448	.438	∞	.363
sonar	∞	.883	∞	.880	1.179	.865	∞	.848
spect	∞	.528	∞	.523	.565	.523	∞	.522
spectf	∞	.719	∞	.711	.964	.710	∞	.692
transf.	∞	.804	∞	.798	.857	.792	∞	.781
ttt	∞	.361	∞	.314	.254	.342	∞	.287
wbc	∞	.407	∞	.398	.356	.383	∞	.372
vote	∞	.565	∞	.524	.496	.540	∞	.491
Mean	∞	**.725**	∞	**.710**	**.795**	**.707**	∞	**.681**
Mean rank	**3.54**	**1.96**	**3.42**	**1.08**	**2.80**	**2.12**	**4.00**	**1.08**

There are several interesting observations that can be made when looking at the Brier losses in Table 3 below. First, we see that using calibration is always clearly better than not. Second, it is obvious that using the Laplace estimates for the calibration is significantly better than using the relative frequencies. Finally, when comparing the three calibration techniques, VAP is actually significantly better than Platt and Iso, both when using relative frequencies and Laplace estimates for the calibration.

Table 5: Brier loss

	RF				LaP			
	Tree	Platt	Iso	VAP	Tree	Platt	Iso	VAP
colic	.176	.155	.148	.147	.151	.145	.133	.132
creditA	.144	.134	.129	.128	.125	.126	.115	.114
diabetes	.244	.197	.194	.193	.208	.191	.182	.181
german	.296	.210	.210	.210	.256	.209	.209	.208
haber	.264	.203	.205	.203	.229	.202	.204	.202
heartC	.215	.183	.180	.178	.188	.175	.170	.168
heartH	.193	.172	.168	.166	.170	.165	.159	.156
heartS	.198	.183	.182	.180	.175	.176	.174	.171
hepati	.201	.153	.157	.151	.176	.151	.156	.149
iono	.111	.104	.101	.100	.097	.098	.091	.091
je4042	.248	.212	.209	.206	.213	.204	.196	.193
je4243	.320	.237	.237	.236	.275	.234	.233	.230
kc1	.269	.190	.190	.190	.233	.189	.188	.188
kc2	.234	.176	.171	.170	.198	.170	.158	.156
kc3	.147	.111	.110	.109	.127	.110	.109	.107
liver	.295	.234	.235	.234	.256	.231	.232	.229
pc1req	.258	.238	.244	.235	.235	.237	.242	.233
pc4	.112	.093	.088	.088	.096	.090	.081	.081
sonar	.268	.210	.211	.209	.234	.204	.202	.199
spect	.128	.102	.103	.103	.115	.101	.104	.104
spectf	.232	.159	.160	.158	.201	.158	.156	.154
transf.	.230	.186	.185	.184	.200	.183	.182	.181
ttt	.056	.070	.064	.063	.053	.068	.060	.060
wbc	.071	.076	.075	.074	.066	.072	.071	.070
vote	.116	.120	.113	.112	.108	.115	.108	.106
Mean	**.201**	**.164**	**.163**	**.161**	**.175**	**.160**	**.157**	**.155**
Mean rank	**3.72**	**2.68**	**2.44**	**1.16**	**3.52**	**3.04**	**2.28**	**1.16**

Table 6: Reliability of estimates

	RF				LaP			
	Tree	Platt	Iso	VAP	Tree	Platt	Iso	VAP
colic	.0497	.0050	.0038	.0031	.0318	.0085	.0038	.0036
creditA	.0240	.0025	.0011	.0010	.0192	.0060	.0012	.0010
diabetes	.0564	.0024	.0017	.0013	.0367	.0036	.0013	.0007
german	.0903	.0013	.0024	.0022	.0510	.0012	.0031	.0025
haber	.0748	.0040	.0074	.0058	.0404	.0033	.0077	.0054
heartC	.0486	.0029	.0037	.0023	.0343	.0025	.0026	.0014
heartH	.0419	.0025	.0034	.0021	.0300	.0023	.0033	.0019
heartS	.0404	.0027	.0045	.0028	.0245	.0014	.0038	.0021
hepati	.0586	.0060	.0104	.0057	.0382	.0049	.0099	.0050
iono	.0154	.0024	.0020	.0016	.0139	.0036	.0015	.0015
je4042	.0551	.0035	.0045	.0026	.0380	.0029	.0038	.0020
je4243	.0900	.0020	.0053	.0038	.0512	.0014	.0053	.0027
kc1	.0822	.0009	.0015	.0014	.0476	.0004	.0017	.0012
kc2	.0727	.0025	.0033	.0022	.0494	.0018	.0030	.0018
kc3	.0454	.0025	.0036	.0029	.0299	.0025	.0042	.0033
liver	.0787	.0028	.0055	.0039	.0455	.0019	.0055	.0032
pc1req	.0601	.0068	.0144	.0058	.0364	.0063	.0143	.0062
pc4	.0284	.0024	.0007	.0006	.0183	.0041	.0008	.0005
sonar	.0697	.0040	.0060	.0040	.0501	.0024	.0047	.0025
spect	.0378	.0047	.0059	.0061	.0250	.0042	.0063	.0066
spectf	.0807	.0037	.0053	.0042	.0539	.0031	.0053	.0037
transf.	.0528	.0024	.0033	.0026	.0294	.0016	.0035	.0021
ttt	.0083	.0033	.0009	.0009	.0064	.0045	.0009	.0011
wbc	.0065	.0022	.0014	.0014	.0059	.0026	.0011	.0012
vote	.0220	.0036	.0026	.0020	.0159	.0051	.0027	.0022
Mean	**.0516**	**.0032**	**.0042**	**.0029**	**.0329**	**.0033**	**.0041**	**.0026**
Mean rank	**4.00**	**1.92**	**2.60**	**1.48**	**4.00**	**1.84**	**2.56**	**1.60**

Table 6 below shows the reliability scores for the different techniques. Again the most obvious result is that using no calibration is a very poor alternative. In fact, it is (often by a large margin) the worst option on each and every data set. Regarding statistically significant differences, all three setups using calibration were of course significantly more reliable than the Tree, both when using relative frequencies and Laplace estimates. In addition, when using relative frequencies, VAP was significantly better than Iso. Using Laplace estimates, both VAP and Platt were significantly better than Iso. To summarize, the results show that all three general calibration methods, i.e., Platt scaling, isotonic regression and Venn-Abers predictors generally improved on the estimates from regular PETs, thus strongly indicating that these kind of techniques may be necessary for converting standard decision trees into truly well-calibrated PETs. Another interesting result is that the estimates from Platt scaling, isotonic regression and Venn-Abers predictors clearly improved when using Laplace estimates instead of relative frequencies for the calibration.

Comparing the different calibration methods, VAP obtained the lowest log and Brier losses, and also the best reliability measure. A deepened analysis showed, as presented above, that most of these differences were statistically significant. In addition, Venn-Abers also produced more accurate PETs, with higher AUC-vales, than both Platt scaling and isotonic regression.

Table 7 below, finally, shows the intervals obtained by the VAP, together with the observed accuracies. As can be expected, there are a few data sets where the empirical accuracy is actually outside of the interval (marked in bold), but of course the overall picture is that the VAPs are well-calibrated. It should also be noted that the intervals are fairly tight, on average smaller than 0.045, but that the sizes vary considerably over the data sets.

Table 7: VAP intervals

	RF			LaP		
	Low	High	Acc	Low	High	Acc
colic	.798	.833	.818	.811	.853	.834
creditA	.824	.849	.834	.827	.861	.838
diabetes	.714	.737	**.712**	.717	.749	.719
german	.694	.709	.701	.692	.711	.700
haberman	.699	.742	.701	.696	.748	.703
heartC	.741	.788	.761	.745	.803	.764
heartH	.759	.806	.768	.763	.819	.775
heartS	.741	.792	.757	.743	.806	.758
hepati	.777	.839	.784	.776	.846	.788
iono	.863	.894	.880	.859	.906	.883
je4042	.688	.740	.695	.695	.757	.703
je4243	.624	.662	**.613**	.626	.675	**.616**
kc1	.729	.742	.730	.733	.753	.737
kc2	.748	.784	.747	.757	.805	.767
kc3	.848	.883	.862	.842	.886	.861
liver	.635	.676	**.626**	.641	.691	**.639**
pc1req	.617	.723	.626	.621	.730	.635
pc4	.873	.887	.874	.876	.895	.882
sonar	.691	.737	.705	.697	.763	.707
spect	.854	.901	.884	.849	.901	.884
spectf	.778	.813	.783	.774	.823	.786
transfusion	.725	.757	.727	.725	.765	.733
ttt	.907	.927	.918	.894	.929	.919
wbc	.895	.926	.914	.892	.931	.915
vote	.832	.870	.844	.831	.873	.846

6 Concluding remarks

This study has presented a large-scale comparison of Venn-Abers predictors to existing techniques for calibrating probabilistic predictions. The empirical investigation clearly showed the capabilities of Venn-Abers predictors; prediction intervals were very tight and the probability estimates extremely well-calibrated. In fact, comparing Venn-Abers estimates to Platt scaling and isotonic regression, using a number of metrics, Venn-Abers was always the most exact. In addition, the empirical study gave strong evidence for the benefit of using external calibration methods when producing well-calibrated PETs. Finally, the results also showed it to be favorable for all three calibration techniques, i.e., Venn-Abers, Platt scaling and isotonic regression, to use Laplace estimates instead of relative frequencies.

References

[1] Andrews, R., Diederich, J., Tickle, A.B.: Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowl.-Based Syst. **8**(6), 373–389 (1995)

[2] Bache, K., Lichman, M.: UCI machine learning repository (2013)

[3] Bergmann, B., Hommel, G.: Improvements of general multiple test procedures for redundant systems of hypotheses. In: Multiple Hypotheses Testing, pp. 100–115. Springer (1988)

[4] Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. Chapman & Hall/CRC (1984)

[5] Freitas, A.A.: A survey of evolutionary algorithms for data mining and knowledge discovery. In: Advances in Evolutionary Computation. Springer (2002)

[6] Freitas, A.A.: Comprehensible classification models: A position paper. SIGKDD Exp. Newsl. **15**(1), 1–10 (2014)

[7] Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of American Statistical Association **32**, 675–701 (1937)

[8] Gammerman, A., Vovk, V., Vapnik, V.: Learning by transduction. In: Uncertainty in artificial intelligence, pp. 148–155. Morgan Kaufmann Publishers Inc. (1998)

[9] Johansson, U., Löfström, T., Linusson, H., Boström, H.: Efficient venn predictors using random forests. Machine Learning (2018)

[10] Johansson, U., Löfström, T., Sundell, H., Linusson, H., Gidenstam, A., Boström, H.: Venn predictors for well-calibrated probability estimation trees. In: COPA, pp. 1–12. PMLR (2018)

[11] Lambrou, A., Nouretdinov, I., Papadopoulos, H.: Inductive venn prediction. Annals of Mathematics and Artificial Intelligence **74**(1), 181–201 (2015)

[12] Murphy, A.H.: A new vector partition of the probability score. Journal of Applied Meteorology **12**(4), 595–600 (1973)

[13] Niculescu-Mizil, A., Caruana, R.: Predicting good probabilities with supervised learning. In: ICML, pp. 625–632. ACM (2005)

[14] 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 (1999)

[15] Provost, F., Domingos, P.: Tree induction for probability-based ranking. Mach. Learn. **52**(3), 199–215 (2003)

[16] Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann (1993)

[17] Sayyad Shirabad, J., Menzies, T.: The PROMISE Repository of Software Engineering Databases. School of IT and Engineering, Univ. of Ottawa, Canada (2005)

[18] Vovk, V., Gammerman, A., Shafer, G.: Algorithmic Learning in a Random World. Springer-Verlag New York, Inc. (2005)

[19] Vovk, V., Petej, I.: Venn-abers predictors. arXiv preprint arXiv:1211.0025 (2012)

[20] Vovk, V., Shafer, G., Nouretdinov, I.: Self-calibrating probability forecasting. In: Advances in Neural Information Processing Systems, pp. 1133–1140 (2004)

[21] Zadrozny, B., Elkan, C.: Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In: ICML, pp. 609–616 (2001)