

# Improving the Reliability of Decision Tree and Naive Bayes Learners

David Lindsay,  
Computer Learning Research Centre,  
Royal Holloway, University of London,  
Egham, Surrey, TW20 OEX, UK,  
davidl@cs.rhul.ac.uk

Siân Cox  
School of Biological Sciences,  
Royal Holloway, University of London,  
Egham, Surrey, TW20 OEX, UK  
s.s.e.cox@rhul.ac.uk

## Abstract

*The C4.5 Decision Tree and Naive Bayes learners are known to produce unreliable probability forecasts. We have used simple Binning [11] and Laplace Transform [2] techniques to improve the reliability of these learners and compare their effectiveness with that of the newly developed Venn Probability Machine (VPM) meta-learner [9]. We assess improvements in reliability using loss functions, Receiver Operator Characteristic (ROC) curves and Empirical Reliability Curves (ERC). The VPM outperforms the simple techniques to improve reliability, although at the cost of increased computational intensity and slight increase in error rate. These trade-offs are discussed.*

## 1. Introduction

Probability forecasting is a generalisation of the standard pattern recognition problem. Rather than attempting to find the “best” label, the aim is to estimate the conditional probability (otherwise known as a probability forecast) of a possible label given an observed object.

The problem of making *effective* probability forecasts is a well studied problem [3][4][7]. Dawid (1985) gives two simple criteria for describing how effective probability forecasts are:

1) **Reliability** - The probability forecasts “should not lie”. When a probability  $\hat{p}$  is assigned to an event, there should be roughly  $1 - \hat{p}$  relative frequency of the event not occurring. This is also referred to as being *well-calibrated*.

2) **Resolution** - The probability forecasts should be practically useful and enable the observer to easily rank the events in order of their likelihood of occurring. This criterion is more related to classification accuracy.

Investigations to improve the reliability of probability forecasts output by popular machine learners such as Naive Bayes and Decision Tree have been considered [11][6]. In this study, we investigate the effectiveness of two simple

approaches to improve reliability, namely the Laplace transform [2] for the C4.5 learner and re-calibration using ‘binning’ [11] for the Naive Bayes and C4.5 learners. We compare the effectiveness of these simple techniques to that of the recently developed Venn Probability Machine (VPM) meta-learner [9]. Unlike the simpler approaches the VPM has proven ability to produce reliable probability forecasts [9]. This study describes the first-ever implementation of the VPM on top of the Naive Bayes and Decision Tree learners and we show that the VPM outperforms simpler techniques to improve reliability. However this comes at a cost of increased computational intensity and a slight increase in error rate. We discuss the implications of these trade-offs and the practical advantages of reliable probability forecasts.

## 2. Reliability: a Machine Learning Perspective

Our notation will extend upon the commonly used supervised learning approach to pattern recognition. Nature outputs information pairs called *examples*. Each example  $(\mathbf{x}_i, y_i)$  consists of an *object*  $\mathbf{x}_i$  and its *label*  $y_i \in \mathbf{Y} = \{1, 2, \dots, |\mathbf{Y}|\}$ . Adapting formulation as in [4], let us consider a sequence of  $n$  probability forecasts for the  $|\mathbf{Y}|$  possible labels output by a learner  $\Gamma$ . Let  $\hat{P}(y_i = j | \mathbf{x}_i) = \hat{p}_{i,j}$  represent the estimated conditional probability of the  $j$ th label matching the true label for the  $i$ th object tested. To calculate reliability for a finite number of forecasts a method of discretising probability forecasts must be used. For predicted probabilities  $\hat{p}_{i,j}$  of each class  $j \in \mathbf{Y}$  we define a set of ‘bins’ (disjoint sub-intervals)  $B_j$ , for example one possible bin choice would be to choose  $k$  equal bins  $B_j = \left\{ \left[0, \frac{1}{k}\right), \left[\frac{1}{k}, \frac{2}{k}\right), \dots, \left[\frac{k-1}{k}, 1\right] \right\}$ . Of course a learner’s probability forecasts are rarely uniformly distributed and so equal width intervals may not be sufficient<sup>1</sup>. Let

<sup>1</sup>For our ERC [6], VPM [9] and binning [11] meta-learner implementations, we used the Discretize filter provided by WEKA [10] which uses an MDL criterion to optimally define bin interval sizes [5].

$n_b^j = \sum_{i=1}^n \mathbb{I}_{\{\hat{p}_{i,j} \in b\}}$  count the number of forecasts  $\hat{p}_{i,j}$  for class  $j \in \mathbf{Y}$  that fall within bin interval  $b \in B_j$ . There are many possible choices of bin sizes, however we aim to specify bin sizes which encompass enough forecasts (make  $n_b^j$  as large as possible for each bin) to obtain practically useful estimates.

Once the sets of bins  $B_j, j \in \mathbf{Y}$  have been defined, we can define reliability by calculating various statistics from the individual bins  $b \in B_j$ . Reliability ensures that for each bin of forecasts with predicted values  $\approx \hat{p}$ , the frequency of this label not occurring in that bin is  $\approx 1 - \hat{p}$ . To obtain a practically useful estimate of the predicted value represented by each bin we use the average predicted probability  $\phi_n^j(b) = \frac{\sum_{i=1}^n \mathbb{I}_{\{\hat{p}_{i,j} \in b\}} \hat{p}_{i,j}}{n_b^j}$  for each bin interval  $b$ . The empirical frequency  $\rho_n^j(b) = \frac{\sum_{i=1}^n \mathbb{I}_{\{y_i=j\}} \mathbb{I}_{\{\hat{p}_{i,j} \in b\}}}{n_b^j}$  of each bin  $b$  calculates the proportion of predictions in that bin that had true class  $y_i = j$ . To determine whether a bin  $b$  contains enough forecasts to be practically useful to gather the  $\rho_n^j(b)$  and  $\phi_n^j(b)$  statistics an extra weighting term  $\nu_n^j(b) = \frac{n_b^j}{n}$  is used. Every learner’s performance in calibration criteria can be categorised using the above functions. Using them intuitively defines reliability. A learner is *well calibrated* (reliable) if its forecasts  $\{\hat{p}_{1,1}, \dots, \hat{p}_{1,|\mathbf{Y}|}, \dots, \hat{p}_{n,1}, \dots, \hat{p}_{n,|\mathbf{Y}|}\}$  and a fixed specification of bins  $B_j, j \in \mathbf{Y}$  satisfy  $R(\Gamma, n) = \sum_{j=1}^{|\mathbf{Y}|} \sum_{b \in B_j} \nu_n^j(b) |\rho_n^j(b) - \phi_n^j(b)| \approx 0$ .

### 3. Methods for Assessing Reliability

At present the most popular techniques for assessing the quality of probability forecasts are *square loss* [10], *log loss* [10] and *ROC curves* [8]. The problem of defining effective scoring rules or loss functions for evaluating probability forecasts has been considered in depth [4]. Each loss function assesses a combination of reliability and resolution, with different biases on the individual components [7]. The area under the ROC curve is commonly used as a measure of the usefulness of the probability forecasts; the larger the area, the better the forecasts.

The *Empirical Reliability Curve* (ERC) is a visual interpretation of the theoretical definition of reliability [6]. Unlike the previous methods, the ERC allows visualisation of over- and under-estimation of probability forecasts. For more detail about ERC implementation please refer to [6]. In brief, each coordinate (marked as a cross) on the ERC represents the statistics computed for each bin  $b$ , and the cross is coloured according the weighting of that bin  $\nu(b)$  (black = 1, 1 > shades of grey > 0, white = 0). A reliable classifier will have ERC coordinates  $(\phi(b), \rho(b))$  close to the diagonal line of calibration  $(0, 0) \rightarrow (1, 1)$  (where predicted probability equals empirical frequency). A trend

line is predicted from these coordinates using a weighted regression algorithm [1] (each training example weighted according to the value  $\nu(b)$ ). This allows the coordinates which relate to a bin containing a large sample of forecasts to have a greater influence on the shape of the curve.

### 4. The Venn Probability Machine (VPM)

The Venn Probability Machine (VPM) framework was designed to complement predictions made by traditional learning algorithms with provably reliable probability forecast bounds in the online setting (where data is continually updated) [9]. However, there is much empirical evidence (as given in this study) to support the fact that these bounds are also reliable for the offline learning setting. The VPM “*sits on top*” of existing learners and can be easily modified from generating bounds to useful point estimates. Essential to the working of the VPM is the  $|\mathbf{Y}| \times |\mathbf{Y}|$  dimensional *Venn probability matrix*  $\mathbf{M}$ .

The VPM is self calibrated by the definition of a fixed method for grouping examples  $z_i = (\mathbf{x}_i, y_i)$  into ‘*types*’. The intuition behind this is that a reasonable statistical forecast would take into account *only* objects  $\mathbf{x}_i$  which are *similar* to the object of interest to obtain reliable estimates. We choose a finite set of types  $\{\lambda_1, \dots, \lambda_{|\Lambda|}\} = \Lambda$  to cluster examples into apriori. We assign types to each example using a *type defining function*  $T_n : \mathbf{Z}^n \rightarrow \Lambda^n$ , with the requirement that the function is *invariant* (i.e. the order in which the training examples are presented to the learning machine does not affect the resulting types assigned for each example). In practical implementations the underlying learning algorithm is used to define the types. For the implementations of VPM Naive Bayes and VPM Decision Tree we take the underlying learners’ probability forecasts and discretise them using the same MDL criterion [5] used in the construction of the ERC plots. These discretised forecasts act as the types for our VPM learner; with this method the number of types is defined dynamically as the algorithm classifies data.

Of course, there are many different choices of types  $\Lambda$  to cluster examples into, and many different ways of defining these functions  $T_n$ . As soon as we specify the type information, the corresponding VPM is defined automatically in a simple way as detailed below. In particular, the learning component of the VPM always lies in the type definitions [9]. For each new test object  $\mathbf{x}_{n+1}$  we compute the  $n + 1$  types of the  $n$  training examples and the new test object with a tentatively assigned label  $T(z_1, \dots, z_n, (\mathbf{x}_{n+1}, j)) = (t_1, \dots, t_n, t_{n+1})$ . Once we have computed the types for each tentatively assigned class label  $y_{n+1} = j$  of the new test object  $\mathbf{x}_{n+1}$ , we compute a row of the  $|\mathbf{Y}| \times |\mathbf{Y}|$  dimensional Venn probability matrix  $\mathbf{M}$ . The rows of the matrix  $\mathbf{M}$  represent the frequen-

cy count of each class label in the set of training examples which have the *same type* as the new test example

$$M_{ij} = \frac{\sum_{k=1}^{n+1} \mathbb{I}_{\{t_k=t_{n+1}\}} \mathbb{I}_{\{y_k=i\}}}{\sum_{k=1}^{n+1} \mathbb{I}_{\{t_k=t_{n+1}\}}}$$

have the property  $\sum_{i=1}^{|\mathbf{Y}|} M_{ij} = 1$ , which guarantees that the predicted probabilities (averages of each column) will sum to one. To extract conditional probabilities for each possible label  $l \in \mathbf{Y}$  from the VPM learner the average of each column of  $\mathbf{M}$  is calculated  $\hat{P}(y_{n+1} = l | \mathbf{x}_{n+1}) = \frac{\sum_{k=1}^{|\mathbf{Y}|} M_{kl}}{|\mathbf{Y}|}$ .

## 5. Experimental Results

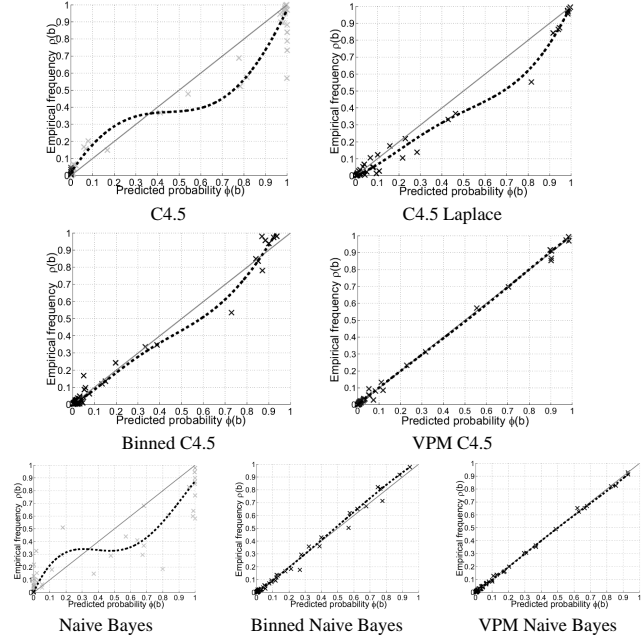
Further details regarding the experimental conditions, datasets and programs used can be found at:

<http://www.clrc.rhul.ac.uk/people/davidl/ICDMResults.html>

In brief, we used 6 real life datasets, five from the UCI data repository and one (Abdominal Pain) from Edinburgh Hospital, UK. All datasets were tested using 10-fold cross validation on an AMD Athlon 2Ghz PC. All programs used are extensions of the WEKA data mining system [10]. Figure 1 shows several Empirical Reliability Curves (ERC) to solely assess the reliability of probability forecasts output by the C4.5 Decision Tree and Naive Bayes learners when tested on the Satellite Image and Abdominal Pain datasets respectively. The solid diagonal line represents the *line of calibration*, where predicted probability (*horizontal*) equals observed empirical frequency of occurrence (*vertical*). Under and over estimation of probability forecasts are represented by the reliability curve (dashed black line) deviating above and below the line of calibration respectively. On their own, both learners display over and under estimation (unreliability) in their probability forecasts. For example, when the Naive Bayes learner makes a prediction with estimated probability 0.9, the empirical frequency of the label occurring is only 0.7 (over estimation). In contrast when a prediction is made with estimated probability 0.2, the empirical frequency of label occurrence is actually  $\approx 0.3$  (under estimation). This phenomenon reiterates the fact that both learners' probability forecasts are too 'extreme'.

Application of the Laplace transform to the C4.5 learner and the Binning technique to both learners improves the reliability of the probability forecasts, as shown by the ERC plots realigning with the ideal line of calibration (Fig. 1). Despite improvement in reliability by the Laplace transform to the C4.5 learner there is still a dramatic over estimation of forecast values  $\hat{p} \approx 0.8$ ; this effect is also observed with the Binning approach, but to a lesser extent. VPM implementations of the Naive Bayes and C4.5 Decision Tree learners produce probability forecasts which are very reliable - as shown by their well-aligned ERC plots (Fig. 1).

We used several assessment scores (Error Rate, ERC deviation area, ROC area, Square & Log loss) to determine the effectiveness of probability forecasts output by the C4.5



**Figure 1. ERC plots visualising reliability of probability forecasts output by various Naive Bayes and C4.5 Decision Tree learners.**

Decision Tree and Naive Bayes learners (and their variants) across six datasets. When scores were ranked, they all ranked differently from the traditionally studied error rate (results not shown), indicating that reliability and classification accuracy do not always go 'hand in hand' [6]. The ERC deviation, ROC area and loss scores supported the improvement in reliability shown by the ERC plots in Fig. 1.

Table 1 gives a summary of the average percentage change (across all six datasets) for each assessment score with respect to the underlying Naive Bayes or C4.5 Decision Tree learner. The VPM C4.5 Decision Tree results indicate a dramatic improvement in terms of the quality of the probability forecasts as compared with the Laplace transform and Binning technique (i.e. ERC deviation decreased by 79% for VPM C4.5 as compared with 12% for C4.5 Laplace and 61% for Binned C4.5). This pattern of improvement is also observed for the Naive Bayes learner, however the difference between the binning approach and VPM is less striking. It is interesting that the ROC area is increased by all techniques applied to the Decision Tree learner yet not for the Naive Bayes learners. This could be because the ROC area may be biased toward measuring the resolution criterion and not reliability [6]. In this instance, the VPM and Binning techniques for the Naive Bayes learner may be forfeiting resolution for reliability. Indeed with VPM implementations of both learners there is on average a slight decrease in classification accuracy. Across all data the VPM is observed to be the slowest in terms of computation

**Table 1. Summary of Results on UCI Datasets**

Average Percentage Change For C4.5 Across Data (%)						
Learner	Error	ERC Dev	ROC Area	Sqr Loss	Log Loss	Time
C4.5 Laplace	0	-12.1	+27.1	-8.4	-61.5	+39.5
Binned C4.5	-0.4	-61.6	+23.1	-10.6	-58.9	+90.6
VPM C4.5	+2.9	-79.2	+32.2	-16.4	-65.6	+1386.5
Average Percentage Change For Naive Bayes Across Data (%)						
Learner	Error	ERC Dev	ROC area	Sqr Loss	Log Loss	Time
Binned Naive Bayes	-6.1	-76.4	-3.2	-15.6	-60.6	+452.6
VPM Naive Bayes	+0.7	-78.6	-4.1	-18.4	-64.1	+672.2

time compared to other techniques tested. This increase in time is most apparent with the VPM Decision Tree with an increase of +1386.5% as opposed to +672.2% with VPM Naive Bayes. This is because the C4.5 Decision Tree is not incremental by design and, unlike the Naive Bayes, when a new example is added to the training data the C4.5 algorithm must be re-run. As indicated earlier, the VPM learner classifies each test example by adding it into the training data with all possible labels tentatively assigned. If the learner’s hypothesis can be cached in memory for the training data and simply updated (as with Naive Bayes) then huge computational shortcuts may be made.

## 6. Discussion

We have tested three methods to improve the reliability of the Naive Bayes and C4.5 Decision Tree learners: Laplace transform, Binning and VPM. The VPM meta-learner outperforms the other techniques in terms of improving reliability, however this is at the cost of slightly increased error rate and increased computation time. Unlike the simpler techniques tested the VPM has the ability to output provably reliable probability forecasts. The C4.5 Decision Tree learner is often limited by small sample sizes settling at leaves of the tree, a problem that is not experienced with VPM as it gains its statistics from the larger sets of the training data (i.e. examples with the same type). The Naive Bayes learner makes simplifying assumptions about the underlying probability distribution (independence of attributes) that rarely hold for real data. In contrast, the VPM learner makes no assumption *further* than the data being *i.i.d.*, which is far more realistic; the only necessary condition for reliability is that the VPM’s type defining function is invariant with respect to the order of the training examples. Indeed another advantage of the VPM approach is that there are many possible type defining functions that could be investigated to yield better results.

One major limitation of the VPM is its computational complexity; if the learner is not naturally incremental, as is the case with the C4.5 Decision Tree, then the computation time can be much larger than for simpler learners, and with large datasets with many possible labels this could be-

come infeasible. It would be interesting therefore to investigate whether incremental versions of Decision Tree learners could be implemented successfully. Another limitation of the VPM is that it can decrease classification accuracy. However we argue that the practical benefit of significantly improving reliability outweighs possible slight increases in error rate. Knowing the reliability of probability forecasts will provide the user with an alternative perspective of a learner’s performance, enabling the user to know whether to ‘trust’ a learner’s predictions [6]. Suppose a user is presented with *differing predictions* from two distinct learners with similar classification accuracy. If reliability is overlooked, the user will probably trust the prediction output by the most accurate learner. However an ERC plot of the probability forecasts made by this learner may reveal gross over estimation. In our view, as practitioners in machine learning we should aim to create learners which are both accurate *and* reliable.

## Acknowledgements

We thank Alex Gammerman, Jo Gates, and Volodya Vovk for their comments. DL was funded by an EPSRC Studentship Grant.

## References

- [1] C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.
- [2] B. Cestnik. Estimating probabilities: A crucial task in machine learning. In *Proc. of the 9th Euro. Conf. on Artificial Intelligence*, 1990.
- [3] A. P. Dawid. Calibration-based empirical probability (with discussion). *Annals of Statistics*, 13:1251–1285, 1985.
- [4] M. H. DeGroot and S. E. Feinberg. The comparison and evaluation of forecasters. *The Statistician*, 32:12–22, 1982.
- [5] U. Fayyad and K. Irani. The attribute selection problem in decision tree generation. In *Proc. of 10th Nat. Conf. on Artificial Intelligence*, 1992.
- [6] D. Lindsay. Visualising and improving reliability - a machine learning perspective. CLRC-TR-04-01, Royal Holloway University of London, England, 2004.
- [7] A. H. Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology*, 12:595–600, 1973.
- [8] F. Provost and T. Fawcett. Analysis and visualisation of classifier performance: Comparison under imprecise class and cost distributions. In *Proc. of the 3rd Int. Conf. on Knowledge Discovery and Data Mining*, 1997.
- [9] V. Vovk, G. Shafer, and I. Nourtdinov. Self-calibrating probability forecasting. In *Advances in Neural Information Processing Systems 16*, 2003.
- [10] I. Witten and E. Frank. *Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco, 2000.
- [11] B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proc. of the 18th Int. Conf. on Machine Learning*, 2001.