VENN PREDICTORS FOR WELL-CALIBRATED PROBABILITY ESTIMATION TREES

Ulf Johansson[1,2], Tuve Löfström[1,2], Håkan Sundell[1,2], Henrik Linusson[2], Anders Gidenstam[2], Henrik Boström[3]

[1]Jönköping University, Sweden.
[2]University of Borås, Sweden.
[3]KTH Royal Institute of Technology, Sweden.

AGENDA

PROBABILISTIC PREDICTION

Introduction

- Many classifiers are able to output not only the predicted class label, but also a probability distribution over the possible classes.
- Naturally, all probabilistic prediction requires that the probability estimates are well-calibrated, i.e., the predicted class probabilities must reflect the true, underlying probabilities.
- If this is not the case, the probabilistic predictions actually become misleading.

Calibration

- In probabilistic prediction, the task is to predict the probability distribution of the label, given the training set and the test object.
- The goal is to obtain a valid predictor.
- In general, validity means that the probability distributions from the predictor must perform well against statistical tests based on subsequent observation of the labels.
- We are interested in calibration: $p(c_j \mid p^{c_j}) = p^{c_j}$, where p^{c_j} is the probability estimate for class j.

PROBABILITY ESTIMATION TREES

PROBABILITY ESTIMATION TREES

- Decision trees are relatively accurate, produce comprehensible models and require a minimum of parameter tuning.
- The two most notable decision tree algorithms are C4.5/C5.0[1] and CART[2].
- Decision trees are readily available for producing class membership probabilities; in which case they are referred to as Probability Estimation Trees (PETs)[3].
- For PETs, the most straightforward way to obtain a class probability 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.
- Intuitively, a leaf containing many training instances is a better estimator of class membership probabilities, so often, a Laplace estimate is used instead.

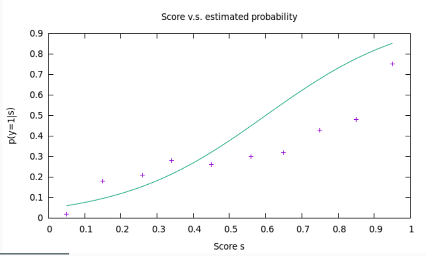
[1] J. R. Quinlan, C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., 1993
[2] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, Classification and Regression Trees, 1984
[3] F. Provost and P. Domingos, "Tree induction for probability-based ranking," Mach. Learn., vol. 52, no. 3, pp. 199–215, 2003

EXISTING APPROACHES FOR CALIBRATION

Platt scaling[4] was originally introduced as a method for calibrating support-vector machines. It works by finding the parameters of a sigmoid function maximizing the likelihood of a calibration set. The function is

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

where $\hat{p}(c \mid s)$ gives the probability that an example belongs to class c, given that it has obtained the score s, and where A and B are parameters of the function found by gradient descent search.

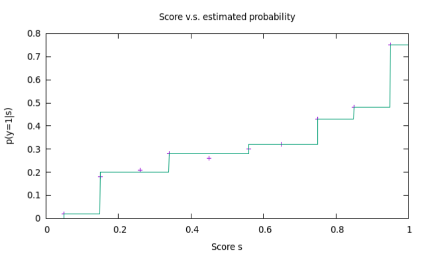


[4] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in Advances in Large Margin Classifiers. MIT Press, 1999, pp. 61–74

Isotonic regression[5] is a calibration method that can be regarded as a general form of binning, not requiring a predetermined number of bins.

The calibration function, which is assumed to be isotonic, i.e., non-decreasing, is a step-wise regression function, which can be learned by an algorithm known as the pair-adjacent violators algorithm.

The algorithm outputs a function that for each input probability interval returns the fraction of positive examples in the calibration set in that interval.



[5] B. Zadrozny and C. Elkan, “Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers,” in Proc. 18th International Conference on Machine Learning, 2001, pp. 609–616

VENN PREDICTORS

Venn predictors[6], are multi-probabilistic predictors with proven validity properties.

Venn predictors was originally suggested in a transductive setting, but here we use the inductive variant:

To construct an inductive Venn predictor, the available labeled training examples $(\{(x_1, y_1), \dots, (x_l, y_l)\})$ are split into two parts, the proper training set $(\{(x_1, y_1), \dots, (x_q, y_q)\})$, used to train an underlying model, and a calibration set $(\{(x_{q+1}, y_{q+1}), \dots, (x_l, y_l)\})$ used to estimate label probabilities for each new test example.

When presented with a new test object x_{l+1}, the aim of Venn prediction is to estimate the probability that $y_{l+1} = Y_j$, for each Y_j in the set of possible labels $Y_j \in \{Y_1, \dots, Y_c\}$.

[6] V. Vovk, G. Shafer, and I. Nouretdinov, "Self-calibrating probability forecasting," in Advances in Neural Information Processing Systems, 2004, pp. 1133–1140

The key idea of Venn prediction is to divide all calibration examples into a number of k categories and use the relative frequency of label $Y_j \in \{Y_1, \ldots, Y_c\}$ in each category to estimate label probabilities for test instances falling into that category.

The categories are defined using a Venn taxonomy and every taxonomy leads to a different Venn predictor.

Typically, the taxonomy is based on the underlying model, trained on the proper training set, and for each calibration and test object x_i, the output of this model is used to assign (x_i, y_i) into one of the categories.

One basic Venn taxonomy, which can be used with every kind of classification model, simply puts all examples predicted with the same label into the same category.

For test instances, the category 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 estimated label probabilities.

As in conformal prediction, 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.

Instead of dealing directly with these distributions, the lower $L(Y_j)$ and upper $U(Y_j)$ probability estimates for each label Y_j are often used.

Let k be the category assigned to the test object x_{l+1} by the Venn taxonomy, and Z_k be the set of calibration instances belonging to category k. Then the lower and upper probability estimates are defined by:

$$L(Y_j) = \frac{\left|\{(x_m, y_m) \in Z_k \mid y_m = Y_j\}\right|}{|Z_k| + 1} \tag{2}$$

and:

$$U(Y_j) = \frac{\left|\{(x_m, y_m) \in Z_k \mid y_m = Y_j\}\right| + 1}{|Z_k| + 1} \tag{3}$$

In order to make a prediction $\hat{y}_{l+1}$ for x_{l+1} using the lower and upper probability estimates, the following procedure is employed in this study:

$$\hat{y}_{l+1} = \max_{Y_j \in \{Y_1, \dots, Y_c\}} L(Y_j) \tag{4}$$

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

$$[L(\hat{y}_{l+1}), U(\hat{y}_{l+1})] \tag{5}$$

METHOD

In the empirical investigation, we look at different ways of producing probability estimates from standard decision trees.

The quality of the probability estimates was measured using the reliability term of the Brier score[7], which is defined as:

$$\frac{1}{N}\sum_{k=1}^{K} n_k(r_k - \phi_k)^2, \tag{6}$$

where, for the interval k, n_k is the number of instances, r_k is the mean probability estimate for the positive class and ϕ_k is the proportion of instances actually belonging to the positive class. We used $K = 100$ intervals.

All experiments were performed in MatLab, so the decision trees were induced using the MatLab version of CART. All parameter values were left at their default values, leading to fairly large trees. Laplace estimates from the trees were used instead of the relative frequencies in all cases.

[7] G. Brier, “Verification of forecasts expressed in terms of probability,” Monthly Weather Review, vol. 78, no. 1, pp. 1–3, 1950

The 22 data sets used are all two-class problems, publicly available from either the UCI repository[8] or the PROMISE Software Engineering Repository[9].

Setups compared

- **LaP**: The Laplace 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 based on the calibration set, where an additional Laplace smoothening was applied to the resulting probability estimates.
- **Venn**: A Venn predictor using a taxonomy where the category is the predicted label from the underlying model, i.e. only two categories are used.

All three methods employing calibration used 2/3 of the training instances for the tree induction and 1/3 for the calibration. Standard 10x10-fold cross-validation were used, so results are averaged over the 100 folds.

[8] Kevin Bache and Moshe Lichman, "UCI Machine Learning Repository," 2013
[9] Sayyad Shirabad, J. and Menzies, T.J., "The PROMISE repository of software engineering databases." 2005

RESULTS

RESULTS – VENN PREDICTOR INTERVALS AND ACCURACY

Data set	Low	High	Size	Accuracy	Data set	Low	High	Size	Accuracy
colic	.777	.795	.019	.790	kc2	.741	.759	.018	.732
creditA	.821	.831	.010	.827	kc3	.857	.878	.021	.867
diabetes	.701	.709	.009	.703	liver	.622	.642	.019	.618
german	.700	.707	.007	.704	mw	.907	.925	.018	.919
haberman	.708	.731	.023	.716	pc4	.872	.877	.005	.869
heartC	.736	.758	.022	.750	sonar	.681	.713	.032	.697
heartH	.748	.771	.023	.760	spect	.867	.896	.029	.886
heartS	.735	.760	.024	.748	spectf	.778	.803	.025	.786
hepati	.781	.824	.043	.789	tic-tac-toe	.905	.912	.007	.910
iono	.858	.877	.019	.877	wbc	.898	.912	.014	.910
kc1	.732	.738	.006	.735	vote	.828	.841	.013	.838

	Estimates				Accuracies				Differences			
Data set	LaP	Platt	Iso	Venn	LaP	Platt	Iso	Venn	LaP	Platt	Iso	Venn
colic	.897	.819	.822	.786	.784	.799	.837	.790	.113	.020	-.015	-.004
creditA	.912	.850	.834	.826	.828	.827	.836	.827	.084	.023	-.002	-.001
diabetes	.872	.733	.726	.705	.712	.715	.720	.703	.160	.017	.006	.002
german	.793	.704	.699	.703	.612	.703	.700	.704	.181	.001	-.001	-.001
haberman	.805	.725	.712	.719	.667	.712	.703	.716	.138	.013	.010	.004
heartC	.876	.773	.761	.747	.734	.753	.757	.750	.142	.020	.004	-.003
heartH	.875	.789	.779	.759	.767	.767	.775	.760	.109	.022	.004	-.001
heartS	.877	.773	.761	.747	.759	.753	.756	.748	.118	.019	.004	-.001
hepati	.893	.820	.794	.802	.772	.793	.784	.789	.121	.027	.010	.013
iono	.941	.889	.867	.867	.880	.879	.884	.877	.061	.010	-.016	-.010
kc1	.858	.737	.740	.735	.683	.735	.736	.735	.176	.002	.004	.000
kc2	.891	.772	.771	.750	.730	.754	.768	.732	.161	.018	.003	.019
kc3	.916	.875	.851	.867	.835	.864	.858	.867	.080	.011	-.007	.000
liver	.827	.646	.659	.632	.639	.632	.641	.618	.188	.014	.018	.014
mw	.936	.924	.902	.916	.897	.916	.914	.919	.039	.007	-.012	-.003
pc4	.945	.889	.880	.874	.871	.879	.881	.869	.074	.010	-.001	.005
sonar	.908	.719	.716	.697	.713	.700	.704	.697	.194	.019	.012	.000
spect	.884	.892	.861	.882	.851	.887	.888	.886	.032	.005	-.027	-.005
spectf	.911	.800	.785	.790	.742	.787	.785	.786	.169	.013	.000	.005
tic-tac-toe	.917	.928	.900	.908	.927	.911	.918	.910	-.010	.017	-.018	-.002
wbc	.941	.922	.899	.905	.915	.911	.916	.910	.026	.011	-.017	-.005
vote	.886	.863	.839	.834	.843	.840	.845	.838	.043	.023	-.006	-.004
Mean	**.889**	**.811**	**.798**	**.793**	**.780**	**.796**	**.800**	**.792**	**.109**	**.015**	**-.002**	**.001**

RESULTS - RELIABILITY OF PROBABILITY ESTIMATES

Data set	LaP	Platt	Iso	Venn
colic	.160	.096	.100	.072
creditA	.179	.126	.128	.104
diabetes	.132	.041	.050	.029
german	.064	.002	.006	.001
haberman	.066	.008	.014	.006
heartC	.152	.080	.081	.063
heartH	.138	.075	.078	.056
heartS	.150	.080	.079	.063
hepati	.090	.029	.031	.022
iono	.186	.136	.126	.117
kc1	.090	.008	.012	.006
kc2	.120	.034	.047	.024
kc3	.057	.010	.016	.007
liver	.111	.020	.026	.015
mw	.036	.007	.011	.005
pc4	.076	.029	.037	.021
sonar	.183	.055	.057	.043
spect	.026	.004	.008	.003
spectf	.105	.015	.022	.012
tic-tac-toe	.172	.165	.152	.144
wbc	.207	.182	.168	.165
vote	.119	.093	.091	.070
Mean	**.119**	**.059**	**.061**	**.048**
Mean Rank	**4.00**	**2.23**	**2.77**	**1.00**

CONCLUSIONS

This paper has presented the first large-scale comparison of Venn predictors to existing techniques for calibrating probabilistic predictions.

The empirical investigation clearly showed the capabilities of a Venn predictor; the produced prediction intervals were very tight, and the probability estimates extremely well-calibrated.

In fact, using the reliability criterion, which directly measures the quality of the probability estimates, the Venn predictor estimates were more exact than Platt scaling and isotonic regression on every data set.

Directions for future work include evaluating Venn prediction as a calibration technique also for other learning algorithms, such as random forests, as well as considering more elaborate approaches for constructing the underlying categories, e.g., by means of so-called Venn-ABERS predictors[10].

[10] V. Vovk and I. Petej, "Venn-abers predictors," arXiv preprint arXiv:1211.0025, 2012

QUESTIONS?

Table: Datasets used in the experiments. #inst denotes the number of instances contained in the dataset; #min and #maj denote the number of examples belonging to the minority and majority classes, respectively. %min is the percentage of examples that belong to the minority class.

Dataset	#inst	#min	#maj	%min	Dataset	#inst	#min	#maj	%min
Colic	357	134	223	37.5	hepatitis	155	32	123	20.6
wbc	699	241	458	34.5	ionosphere	351	126	225	35.9
credit-a	690	307	383	44.5	kc3	325	42	283	12.9
german	1000	300	700	30.0	liver-disorders	345	145	200	42.0
diabetes	768	268	500	34.9	mw	379	30	349	7.9
haberman	306	81	225	26.5	pc4	1343	177	1166	13.1
heart-c	303	138	165	45.5	sonar	208	97	111	46.6
heart-h	294	106	188	36.1	spect	218	24	194	11.0
heart-s	270	120	150	44.4	spectf	267	55	212	20.6
kc1	1192	315	877	26.4	tic-tac-toe	958	332	626	34.7
kc2	369	99	270	26.8	vote	517	144	373	27.8

References

- J. R. Quinlan, C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., 1993.
- L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, Classification and Regression Trees, 1984.
- F. Provost and P. Domingos, "Tree induction for probability-based ranking," Mach. Learn., vol. 52, no. 3, pp. 199–215, 2003.
- J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in Advances in Large Margin Classifiers. MIT Press, 1999, pp. 61–74.
- B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers," in Proc. 18th International Conference on Machine Learning, 2001, pp. 609–616.
- V. Vovk, G. Shafer, and I. Nouretdinov, "Self-calibrating probability forecasting," in Advances in Neural Information Processing Systems, 2004, pp. 1133–1140.
- G. Brier, "Verification of forecasts expressed in terms of probability," Monthly Weather Review, vol. 78, no. 1, pp. 1–3, 1950.

- Kevin Bache and Moshe Lichman, “UCI Machine Learning Repository,” 2013.
- Sayyad Shirabad, J. and Menzies, T.J., “The PROMISE repository of software engineering databases.” 2005.
- V. Vovk and I. Petej, “Venn-abers predictors,” arXiv preprint arXiv:1211.0025, 2012.