Venn–Abers partial ordering method applied to ExCAPE datasets

Ivan Petej

July 31, 2018

Abstract

This brief report outlines the method of Venn–Abers partial ordering method applied to a sample of ExCAPE datasets with some preliminary results.

1 Introduction

Venn predictors produce probability-type predictions for the labels of test objects which are guaranteed to be well calibrated under the standard assumption that the observations are generated independently from the same distribution. Recently a new class of Venn predictors were introduced, called Venn–Abers predictors [4], based on the idea of isotonic regression. with promising empirical results.

One of the drawbacks of Venn-Abers predictors is that their computational efficiency is relatively weak compared with many standard machine learning algorithms. More recently two new versions of Venn–Abers predictors were reported in [5] referred to as inductive Venn–Abers predictors (IVAPs) and cross-Venn–Abers predictors (CVAPs). Just like Venn–Abers predictors, IVAPs and CVAPs automatically enjoy a property of validity (perfect calibration) but are computationally more efficient. The price to pay for perfect calibration is that these probabilistic predictors produce imprecise (in practice, almost precise for large data sets) probabilities. When these imprecise probabilities are merged into precise probabilities, the resulting predictors, while losing the theoretical property of perfect calibration, are shown to be consistently more accurate than the existing methods in empirical studies.

There are several algorithms for performing isotonic regression on a partially, rather than linearly, ordered set: see [1], Section 3.2. The importance of partially ordered

scores stems from the fact that they enable us to benefit from a possible "synergy" between two or more prediction algorithms [3]. Preliminary results reported in [3] in a related context suggest that the resulting predictor can outperform predictors based on the individual scalar scores. An interesting short project would be to test whether a similar result holds for partially ordered IVAP and CVAP equivalents.

This report outlines an initial study into the Venn-Abers partial ordering method applied to a sample of ExCAPE project datasets. The following sections briefly illustrate the method and some preliminary experimental results.

2 Venn–Abers partial ordering method

Consider a standard machine learning problem with two underlying algorithms both generating scores $s_1^i, ..., s_n^i$ for n different examples $(1, ..., n)$ with i representing the index of the algorithm ($i = 1, 2$). In the simplest case we can assume that the scores are calibrated so that they represent probabilities of labels $y_1, ..., y_n$ taking the value of $y = 1$, where set of all labels is binary and each $y \in (0, 1)$. The set of examples therefore consists of $(z_1, \ldots, z_n) = (s_1^1, \ldots, s_n^1; s_1^2, \ldots, s_n^2; y_1, ..., y_n)$.

In general the ordering of the scores for the set of all examples of two algorithms will not be the same - if we order the scores of individual examples by using the first underlying algorithm as a reference (we refer to this transformation as π of $\{1, \ldots, n\}$ such that $z_1, \ldots, z_n \longrightarrow z_{\pi(1)}, \ldots, z_{\pi(n)}$), the resulting scores of the second algorithm will possibly be non-monotonic. This will result in a partial order, with only some of the set of points satisfying $s^1_{\pi(k)} < s^1_{\pi(l)}$ and $s^2_{\pi(k)} < s^2_{\pi(l)}$ simultaneously, where $l > k$. This partial ordering problem can be viewed as a set of simultaneous equations represented as a directed acyclic graph (DAG) with a solution that can be derived using a quadratic optimisation problem as described in [2]. In this report we used the method described in [2] and combined it with the IVAP method in [5] which resulted in Algorithm 1

The method above forms the basis for the experimental work described in the next section.

3 Experimental results

The data consists of 50 splits from the original ExCAPE dataset with scores for three underlying algorithms: k-nearest neighbours (kNN) represented as 1, decison trees (XGB) represented as 2 and support vector machines (SVM) represented as 3 obtained in the

Algorithm 1 Venn–Abers partial ordering algorithm

Input: calibration sequence $(z_1^c, \ldots, z_k^c) = (s_1^{1c}, \ldots, s_k^{1c}; s_1^{2c}, \ldots, s_k^{2c}; y_1^c, \ldots, y_k^c)$ for two underlying algorithms

Input: test sequence $(z_1^t, \ldots, z_n^t) = (s_1^{1t}, \ldots, s_n^{1t}; s_1^{2t}, \ldots, s_n^{2t})$ for two underlying algorithms

Output: probabilistic output $(p_1, \ldots, p_n)$ for $(y_1^t, ..., y_n^t) = 1$ based on Venn–Abers partial ordering of two underlying algorithms

1: **for** $l = 1, \ldots, n$ **do**

2: **for** each $y \in (0, 1)$ **do**

3: Set $(z_1^c, \ldots, z_n^c, z_l^t) = (s_1^{1c}, \ldots, s_k^{1c}, s_l^{1t}; s_1^{2c}, \ldots, s_k^{2c}, s_l^{2t}; y_1^c, \ldots, y_k^c, y)$

4: Derive p_l^y based on the DAG convex optimisation method

5: $p_l = p_l^1/(1 - p_l^0 + p_l^1)$

form of the training, calibration and test sets. All the labels were converted so that $y \in (0, 1)$. The individual scores were first converted into probability outputs using the IVAP method described in [5]. Due to the fact that the first study was run on a conventional PC a subset of 100 calibrating and testing data points were randomly selected for each run of the experiment. For each pair of algorithms out of a possible 3 pairs (1 & 2 , 1 & 3 and 2 & 3) the partial Venn–Abers Algorithm 1 was repeated 5 times. The overall root mean squared error (RMSE) and mean log loss (MLE) as defined in [4] Section 5.6 were compared with the equivalent for each underlying algorithm and the probabilistic prediction based on their average. All calculations were performed in Matlab with the function `quadprog` used fo solving the quadratic optimisation problem. A summary of the results is shown in Table 1.

We can see that the initial results do not look too promising, with the Venn–Abers partial ordering method realising the highest loss in each run. However part of the problem may be in the fact that the data is highly unbalanced with the occurrence of $y = 1$ disproportionally less than the alternative. One attempt at correcting this is by assuming an average label y^k instead of $y = 1$ in steps 2 to 4 of Algorithm 1, given by the average of $1s$ in the calibration set. The experiments were repeated and the overall results are shown in Table 2.

The results are significantly better, with the partial Venn–Abers algorithm outperforming the underlying algorithms and their average in 12 out of 15 runs for RMSE and 11 out of 15 for MLE. The results seem to be least pronounced for a combination of kNN and XGB but significant for other combinations. The lowest overall losses across

Table 1: Root mean square loss (RMSE) and log loss (MLE) results obtained for individual pairs (*first algo*, *second algo*) of three underlying algorithms (k-NN referred to as 1, XGB as 2 and SVC as 3 in the column *Comb*), the algorithm based on an the combined average of a given pair (column *average*) and the Venn–Abers method using partial orders (column *partial*) across different runs. The best results for each pair (algorithm combination, run) are in bold.

	RMSE				MLE			
Comb	*first algo*	*second algo*	*average*	*partial*	*first algo*	*second algo*	*average*	*partial*
1_2	0.022	0.022	**0.020**	0.028	0.070	0.062	**0.062**	0.107
1_2	0.014	0.011	**0.011**	0.021	0.044	**0.037**	0.040	0.076
1_2	0.009	0.014	**0.010**	0.020	0.037	0.037	**0.036**	0.065
1_2	0.033	**0.029**	0.030	0.033	0.111	**0.087**	0.095	0.118
1_2	**0.010**	0.027	0.017	0.035	0.043	0.094	**0.066**	0.132
1_3	0.036	**0.030**	0.031	0.044	0.141	0.114	**0.124**	0.169
1_3	0.034	0.034	**0.033**	0.039	0.117	0.113	**0.111**	0.141
1_3	**0.016**	0.023	0.019	0.027	**0.043**	0.077	0.051	0.096
1_3	0.025	0.024	**0.023**	0.037	0.089	**0.080**	0.082	0.148
1_3	0.024	0.024	**0.023**	0.027	0.080	0.081	**0.078**	0.100
2_3	0.013	0.018	**0.015**	0.021	0.038	0.050	**0.044**	0.061
2_3	0.034	**0.032**	0.032	0.033	0.124	**0.108**	0.115	0.113
2_3	**0.024**	0.035	0.028	0.040	**0.073**	0.114	0.090	0.145
2_3	0.019	**0.015**	0.016	0.019	0.065	**0.056**	0.060	0.057
2_3	0.027	**0.026**	0.026	0.029	0.095	**0.086**	0.089	0.092

all runs are achieved by the Venn–Abers partial ordering algorithm.

Table 2: Root mean square (RMSE) and log loss (MLE) results obtained for individual pairs (*first algo*, *second algo*) of three underlying algorithms (k-NN referred to as 1, XGB as 2 and SVC as 3 in the column *Comb*), the algorithm based on an the combined average of a given pair (column *average*) and the Venn–Abers method using partial orders (column *partial*) across different runs with steps 2 to 4 in Algorithm 1 adjusted to reflect the relative proportion $y = 1$ in the calibration set. The best results for each pair (algorithm combination, run) are in bold.

	RMSE				MLE			
Comb	*first algo*	*second algo*	*average*	*partial*	*first algo*	*second algo*	*average*	*partial*
1_2	0.014	0.011	0.011	**0.002**	0.048	0.042	0.044	**0.005**
1_2	0.033	**0.022**	0.026	0.032	0.123	**0.074**	0.096	0.121
1_2	0.046	**0.041**	0.043	0.043	0.206	**0.174**	0.179	0.213
1_2	0.032	0.032	0.030	**0.028**	0.109	0.117	**0.111**	0.136
1_2	0.013	0.012	0.011	**0.001**	0.045	0.044	0.044	**0.002**
1_3	0.018	0.023	0.018	**0.015**	0.063	0.080	0.070	**0.022**
1_3	0.021	0.015	0.016	**0.003**	0.070	0.051	0.059	**0.005**
1_3	0.027	0.026	0.023	**0.020**	0.089	0.087	0.084	**0.037**
1_3	0.023	0.027	0.023	**0.022**	0.073	0.094	0.077	**0.056**
1_3	0.021	0.022	0.021	**0.019**	0.070	0.074	0.071	**0.033**
2_3	0.025	0.024	0.024	**0.020**	0.087	0.075	0.08	**0.044**
2_3	0.027	0.029	0.028	**0.022**	0.080	0.097	0.086	**0.065**
2_3	0.017	0.020	0.017	**0.001**	0.048	0.053	0.049	**0.001**
2_3	0.030	0.030	0.029	**0.028**	0.111	0.113	0.112	**0.108**
2_3	0.029	**0.027**	0.028	0.028	0.093	**0.079**	0.084	0.118

4 Conclusion

This report described a preliminary study into utilising a modified Venn–Abers method based on partial ordering of two algorithms. Initial results look interesting, however they are based on a small subset of the overall dataset. It may be useful to extend the study further to check the significance of the initial results.

References

[1] Daniel Brunk, Richard Barlow, David Bartholomew, and James Bremner. *Statistical inference under order restrictions.(the theory and application of isotonic regression)*. Tech. rep. University of Columbia, Department of Statistics, 1972.

[2] Rasmus Kyng, Anup Rao, and Sushant Sachdeva. "Fast, provable algorithms for isotonic regression in all l_p-norms". In: *Advances in Neural Information Processing Systems*. 2015, pp. 2719–2727.

[3] Vladimir Vapnik and Rauf Izmailov. "Synergy of monotonic rules". In: *Journal of Machine Learning Research* 17.136 (2016), pp. 1–33.

[4] Vladimir Vovk and Ivan Petej. "Venn-Abers predictors". In: *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*. AUAI Press. 2014, pp. 829–838.

[5] Vladimir Vovk, Ivan Petej, and Valentina Fedorova. "Large-scale probabilistic predictors with and without guarantees of validity". In: *Advances in Neural Information Processing Systems*. 2015, pp. 892–900.