Proceedings of Machine Learning Research 128:1–17, 2020 Conformal and Probabilistic Prediction and Applications

Evaluating Different Approaches to Calibrating Conformal Predictive Systems

Hugo Werner HUGOWER@KTH.SE
School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden
Stena Line, Sweden

Lars Carlsson LARS.CARLSSON@STENALINE.COM
Stena Line, Sweden and Centre for Reliable Machine Learning, University of London, UK

Ernst Ahlberg ERNST.AHLBERG@STENALINE.COM
Stena Line, Sweden and Department of Pharmaceutical Biosciences, Uppsala University, Uppsala, Sweden

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

Editor: Alexander Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov and Giovanni Cherubin

Abstract

Conformal predictive systems (CPSs) provide probability distributions for real-valued labels of test examples, rather than point predictions (as output by regular regression models) or confidence intervals (as output by conformal regressors). The performance of a CPS is dependent on both the underlying model and the way in which the quality of its predictions is estimated; a stronger underlying model and a better quality estimation can significantly improve the performance. Recent studies have shown that conformal regressors that use random forests as the underlying model may benefit from using out-of-bag predictions for the calibration, rather than setting aside a separate calibration set, allowing for more data to be used for training and thereby improving the performance of the underlying model. These studies have furthermore shown that the quality of the individual predictions can be effectively estimated using the variance of the predictions or by k-nearest-neighbor models trained on the prediction errors. It is here investigated whether these methods are also effective in the context of split conformal predictive systems. Results from a large empirical study are presented, using 33 publicly available datasets. The results show that by using either variance or the k-nearest-neighbor method for estimating prediction quality, a significant increase in performance, as measured by the continuous ranked probability score, can be obtained compared to omitting the quality estimation. The results furthermore show that the use of out-of-bag examples for calibration is competitive with the most effective way of splitting training data into a proper training set and a calibration set, without requiring tuning of the calibration set size.

Keywords: Conformal predictive distributions · Split conformal predictive systems · Quality estimation · Regression · Random forest

1. Introduction

Inductive conformal prediction (ICP) (Papadopoulos et al., 2002) was proposed to address the computational cost of the original, transductive, approach to conformal prediction (Gammerman et al., 1998). While the original approach requires re-training for each new test example, ICP requires training of one model only. This however requires that the training set is divided into a proper training set and a calibration set, which reduces the number of examples that are available for model building. Conformal regressors is a class of conformal predictors with a continuous label set. They have been combined with several different machine learning algorithms, e.g., ridge regression (Papadopoulos et al., 2002), neural networks (Papadopoulos and Haralambous, 2010), kNN (Papadopoulos et al., 2011a), and random forests (Johansson et al., 2014; Boström et al., 2017). The benefit of using these algorithms within the conformal regression framework is that the resulting models are valid; the true labels are within the output prediction sets (intervals) with a specified probability (confidence level). The size of a prediction set indicates the degree of uncertainty in the prediction, and also provides information of what labels are unlikely. This has been explored and used in a number of applications, such as in air pollution prediction (Ivina et al., 2012) and in early drug discovery (Svensson et al., 2018).

However, in a decision making context, one typically needs more fine-grained information than just the prediction set; instead, probability distributions over the possible labels may be needed. In the case of classification, the Venn-ABERS predictors (Vovk and Petej, 2012) was developed to generate valid probabilities. Recent developments (Vovk et al., 2017) has introduced probability distributions in the regression setting based on conformal transducers. This approach has been made more computationally efficient in (Vovk et al., 2019), in which split conformal predictive systems were introduced.

In the present work, we investigate whether previous findings on techniques for generating effective conformal regressors carry over to conformal predictive systems. Specifically, we investigate whether the use of a quality estimate, indicating the difficulty on an example-level rather than assuming a uniform quality, leads to more effective conformal predictive systems; using k-nearest neighbors, as have been employed for conformal regressors in e.g., (Papadopoulos and Haralambous, 2011; Johansson et al., 2014), and a computationally more efficient variance-based approach, evaluated in (Boström et al., 2017). Furthermore, we investigate potential performance gains from using out-of-bag predictions for obtaining the calibration scores, rather dividing the training examples into proper training examples and calibration examples, which has been investigated for conformal regressors, when employing random forests as the underlying model, in (Johansson et al., 2014; Boström et al., 2017).

The remainder of this paper is organized as follows; in section 2 we will, for convenience, restate the definition of the split conformal predictive system (Vovk et al., 2019) and also describe the proposed modification, which allows for using all training examples for model construction. Next, in section 3, we will describe how the approach is assessed, followed by

the results from the empirical investigation. Finally, we discuss the results and summarize the main findings in section 4 and section 5, respectively.

2. Methods

2.1. Conformal predictive systems

Conformal predictive systems (CPS) is a modification of conformal predictors. CPS arranges the p-values into a predictive distribution for each example, thus providing probabilities of various outcomes. The CPS used in this paper is the *split conformal predictive systems* (SCPS) which is a computationally efficient method presented in (Vovk et al., 2019). SCPS guarantees validity when the data is i.i.d. In this section, we briefly describe SCPS.

Let $\mathbf{X}$ be the object space and $\mathbf{Z} : \mathbf{X} \times \mathbb{R}$ be the example space such that $(x, y) = z \in \mathbb{Z}$, $y \in \mathbb{R}$. Now we can define the split conformity measure, $A_m : \mathbf{Z}^{m+1} \to \mathbb{R} \cup \{-\infty, \infty\}, m = 1, 2, ...,$ which makes up the core of SCPS. It is defined as a means to compare the size of a new label with previously observed labels. We can now use this to define the SCPS algorithm, see Algorithm 1.

Algorithm 1 Split Conformal Predictive System

Let $\{z_1, ..., z_m\}$ be a training set and $\{z_{m+1}, ..., z_n\}$ be a calibration set.
for $i \in 1, ..., n - m$ **do**
  Define C_i by $A_m(z_1, ..., z_m, z_{m+1}) = A_m(z_1, ..., z_m, (x, C_i))$
end
Sort C_i in increasing order such that $C_{(1)} \leq ... \leq C_{(n-m)}$
Set $C_0 = -\infty$ and $C_{n-m+1} = \infty$
Return the predictive distribution Q.

In Algorithm 1 the predictive distribution Q is defined as,

$$Q(z_1, ..., z_n, (x, y), \tau) := \begin{cases} \frac{i+\tau}{n-m+1} & \text{if } y \in (C_{(i)}, C_{(i+1)}) \text{ for } i \in \{0, 1, ..., n-m\}, \\ \frac{i'+1+(i''-i'+2)\tau}{n-m+1} & \text{if } y = C_{(i)} \text{ for } i \in \{1, ..., n-m\}, \end{cases} \tag{1}$$

where $i' := \min\{j | C_{(j)} = C_{(i)}\}$, $i'' := \max\{j | C_{(j)} = C_{(i)}\}$ and $\tau \sim \mathcal{U}(0, 1)$ is independently sampled for each y_i. Let $\hat{y}$ be a prediction for y and $\hat{\sigma}$ an estimate of the quality of $\hat{y}$. The conformity measure may then be defined as

$$A_m(z_1, ..., z_m, z_{m+1}) := \frac{y - \hat{y}}{\hat{\sigma}}. \tag{2}$$

There are other ways of defining the conformity measure but the above definition makes it computationally efficient to calculate C_i and thus making Algorithm 1 computationally efficient. With the conformity measure defined as (2) and $A_m(z_1, ..., z_m, z_{m+1}) = A_m(z_1, ..., z_m, (x, C_i))$, C_i is defined as

$$C_i := \hat{y} + \frac{\hat{\sigma}}{\hat{\sigma}_{m+i}}(y_{m+1} - \hat{y}_{m+i}). \tag{3}$$

The performance of conformal predictors is often evaluated by the efficiency, i.e., the size of the produced prediction intervals. CPSs may in principle also be evaluated with respect to efficiency, but this would require a significance level to be chosen and would furthermore not fully capture the quality of the predictive distribution generated by CPSs. To evaluate the performance of the predictive distribution, we instead consider the same loss function used in (Vovk et al., 2019), namely continuous ranked probability score (CRPS). If we let $F : \mathbb{R} \to [0, 1]$ be the distribution function and y_i the label, then the CRPS is defined as

$$\text{CRPS}(F, y_i) = \int_{-\infty}^{\infty} (F(y) - \mathbf{1}_{y>y_i})^2 dy. \tag{4}$$

The lower the CRPS, the better the performance, with a lowest possible value of 0. Due to the fuzziness of SCPS, CRPS cannot be computed directly. However, with the following modification of Q, (Vovk et al., 2019), the fuzziness is ignored and using CRPS as a measurement of performance becomes viable,

$$Q(z_1, ..., z_n, (x, y), \tau) := \begin{cases} \frac{i}{n-m} & \text{if } y \in (C_i, C_{i+1}) \text{ for } i \in \{0, 1, ..., n-m\}, \\ \\ \frac{i}{n-m} & \text{if } y = C_i \text{ and } y \neq C_{i+1} \text{ for } i \in \{1, ..., n-m\}. \end{cases} \tag{5}$$

2.2. Underlying model

Random forests are ensembles of random trees (Breiman, 2001). Each random tree is a decision tree trained on a bootstrap replicate of the training examples, i.e., n examples are sampled with replacement from the original n training examples. Thus each tree will be constructed from only a subset of the training set. The examples which are not used during training of a particular tree are said to be out-of-bag (oob) examples for that tree.

SCPS uses a training set to train the underlying model and a calibration set to generate the predictive distribution. With a finite number of examples in a dataset, there is a trade-off between the number of examples to use for training versus calibration. In general, the more examples in the training set, the stronger will the underlying model be. However with fewer examples in the calibration set, the more coarse the distribution will be. As suggested in (Devetyarov and Nouretdinov, 2010) and further investigated in (Johansson et al., 2014), in the context of conformal prediction, an alternative to splitting the dataset is to use the oob predictions of a random forest when obtaining the calibration scores. The advantage of this method is that all examples can be used for both training and calibration. There is however a potential down-side to the method. When predicting the label for an oob example, only the trees that were not trained from that particular example will be used, i.e., only a subset of the trees in the forest are used for each oob example. Hence, the predictions by such subsets of the forest can be expected to be less accurate than when using the entire forest to form the predictions. Moreover the validity is lost, however, (Johansson et al., 2014) and (Boström et al., 2017) show that when using oob predictions within the framework of conformal prediction, validity holds empirically.

By accurately estimating the quality of the predictions of the underlying model, the performance of the SCPS may be improved. In (Boström et al., 2017), two methods for estimating the quality of predictions in the context of conformal regression forests were

investigated. The first method employed the variance of the predictions of the trees in the random forest. The second method, which was first proposed in (Papadopoulos et al., 2011b), used a weighted average of the prediction error of the k nearest neighbours.

2.3. No quality estimation

Conformal predictive distributions can be used without estimating the quality of each prediction. By setting $\hat{\sigma} = 1$ for all examples, i.e., both in the calibration set and the test set, we assume that all predictions will be of equal quality. Thus the characteristics of the predictive distribution will be the same differing only by a shift depending on the prediction of the underlying model. Although an SCPS without quality estimation provides more information than a point prediction, it is of interest to be able to estimate the quality and thus get a unique predictive distribution for each example.

2.4. Variance

The first method used for estimating the quality of predictions is to use the variance of the predictions for the different trees. For easily predicted examples, the trees in the forest can be expected to agree more and hence the variance will be small. On the contrary, for uncertain predictions, the trees will typically differ in their predictions, which results in a larger variance. Assuming that the forest has N trees and that $\{p_1, p_2, ..., p_N\}$ are the predictions of the trees in the forest and that $\beta > 0$ is some small number to avoid division by zero in (3). The quality estimation, $\hat{\sigma}$, for an example using the variance is then defined as

$$\hat{\sigma} = \frac{1}{N}\sum_{t=1}^{N} p_t^2 - \frac{1}{N^2}\left(\sum_{t=1}^{N} p_t\right)^2 + \beta \tag{6}$$

In the case of using the oob method, only a subset of the trees are used for the calibration set. The variance will also be calculated from this subset of trees. To best replicate this for the quality estimation of the test examples, trees are selected by a bootstrap replicate and the variance of the predictions of the trees which are not included in the bootstrap replicate is calculated.

2.5. kNN

The second method used for estimating the quality of predictions is by calculating the average out-of-bag error of the k nearest examples weighted by the Euclidean distance. This is motivated by the assumption that the quality of predictions of examples that are separated by a small Euclidean distance will differ less than those further apart. Let $\{o_1, o_2, ..., o_k\}$ be the oob errors, d_j the Euclidean distance between the (calibration or test) example and its k nearest neighbors and $\beta > 0$ a small number to avoid division by zero in (3). Then $\hat{\sigma}$ is defined as

$$\hat{\sigma} = \frac{\sum_{j=1}^{k} o_j/d_j}{\sum_{j=1}^{k} 1/d_j} + \beta \tag{7}$$

3. Experiments

The experiments were designed to compare the performance of splitting the dataset into a test set and calibration set with the use of the entire dataset for training and using oob predictions for the calibration in the conformal predictive distribution setting. Another purpose of the experiment was to compare and evaluate the two above quality estimation methods in the conformal predictive distribution setting. In these experiments, the performance was measured by CRPS.

3.1. Experimental setup

For the experiments in this paper, the same 33 publicly available datasets were used as considered in (Boström et al., 2017) and (Johansson et al., 2014). Details of these datasets are listed in Table A.1 in the Appendix. The number of instances in the datasets ranges roughly from 500 to 10000 and the number of attributes ranges from 2 to 15. To allow for comparing the results, the labels (y) were normalized by

$$\tilde{y}_i = \frac{y_{\max} - y_i}{y_{\max} - y_{\min}} \tag{8}$$

In the experiments, the different quality estimation methods were employed and for each method, different sizes of the training and calibration sets were considered together with using the entire dataset for training and the oob examples for calibration. The considered ratios of the sizes of the training and calibration sets were 1:9, 2:8, ..., 9:1. In all experiments, 30% of each dataset was used as a test set. The experiments were repeated 10 times with each dataset split randomly into the proper training set, calibration set and test set. The CRPS score for each dataset and experiment was calculated by taking the average of the CRPS of all examples in the test sets. The parameters used were the same for all datasets. Following (Johansson et al., 2014) and (Boström et al., 2017), the parameter β in equation (6) and (7) was set to 0.01 and the number of trees in the forest was set to 500. When the kNN method was used for estimating the quality of the predictions, the 25 nearest neighbors were used, as suggested in (Johansson et al., 2014). To compare the results of the different methods and determine if there was a significant difference, a Friedman test was performed, where the null hypothesis states that there is no difference in performance between the methods. To detect pairwise differences, the Friedman test was followed by a Nemenyi test (Demšar, 2006).

3.2. Experimental results

For each combination of dataset, quality estimation method and training set size, ten CRPS scores were obtained and averaged, yielding a mean CRPS for each combination. Figures 1, 2, and 3 show box plots of the mean CRPS scores. Each figure shows the results from one of the quality estimation methods. For all three methods, the plots show that the more data that is used for training, the lower the average CRPS, i.e., an increase in performance. However, the spread of the results remains large for all sizes of the training and calibration set. The average CRPS scores from the ten runs for all the experiments are included in the Appendix in Tables A.2, A.3 and A.4.

In Fig. 4 and 6, the results from the corresponding Nemenyi tests are shown. The figures show the average rank of the CRPS for the different training set sizes and the oob method. They also show the critical difference (CD) for an $\alpha = 0.05$. The methods which are not connected by a black bar are significantly different from each other. As seen from the box plots, the performance increases with the size of the training set, while the oob method has the best ranking in all three setups. However, there is no significant difference in the performance between using most (80% or 90%) of the available data for training and the rest for calibration compared to using the oob method. In Fig. 7, the performances of the oob methods for each quality estimation method are compared. The plot shows that, again at $\alpha = 0.05$, there is a significant difference between omitting the quality estimation and using either of the employed quality estimation methods; both comparisons yield a p-value of 0.001. There is however no significant difference between the two quality estimation methods.

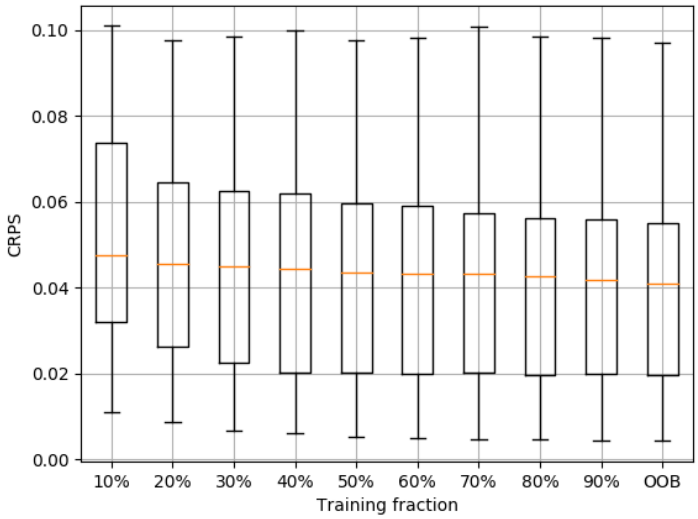

Figure 1: Results for different calibration approaches without quality estimation, i.e., $\hat{\sigma} = 1$.

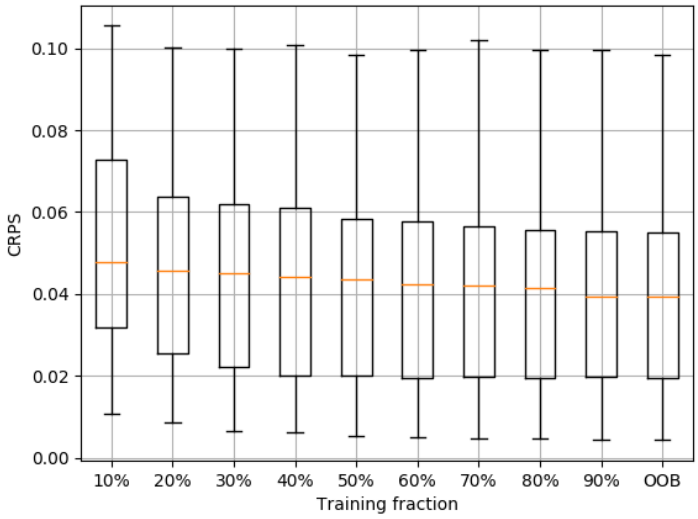

Figure 2: Results for different calibration approaches when estimating quality by variance.

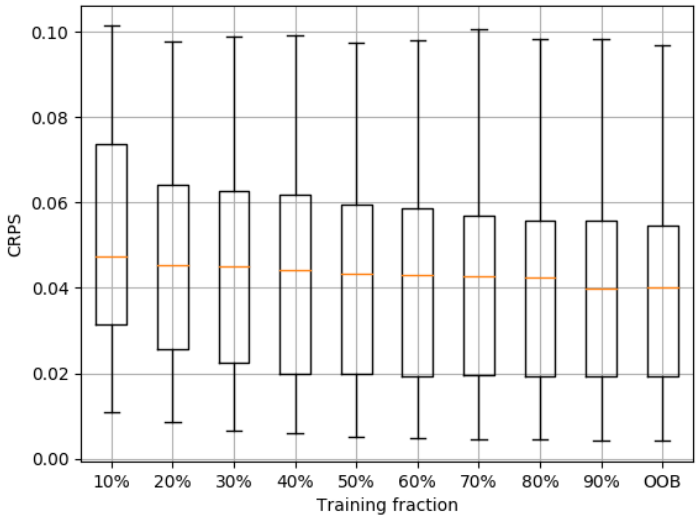

Figure 3: Results for different calibration approaches when estimating quality by kNN.

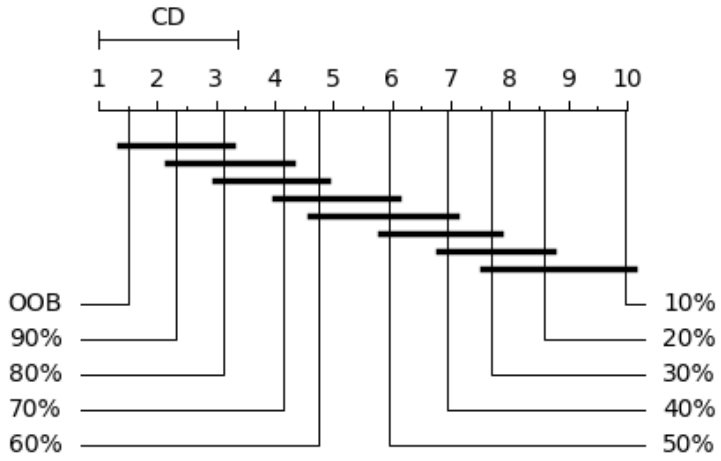

Figure 4: Nemenyi test of the results without quality estimation, i.e., $\hat{\sigma} = 1$.

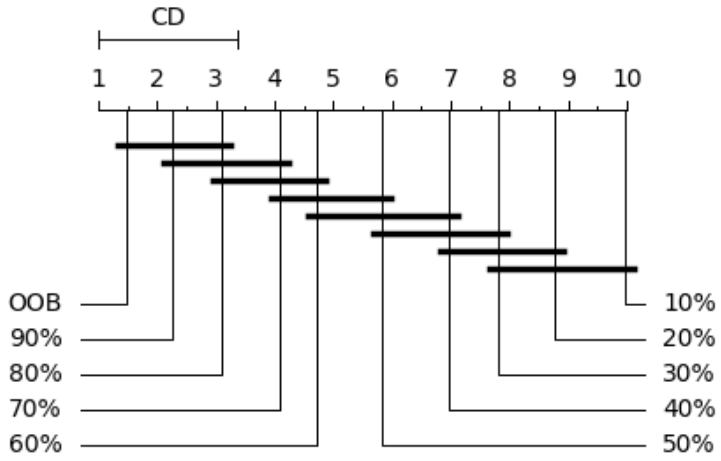

Figure 5: Nemenyi test of the results when estimating quality by variance.

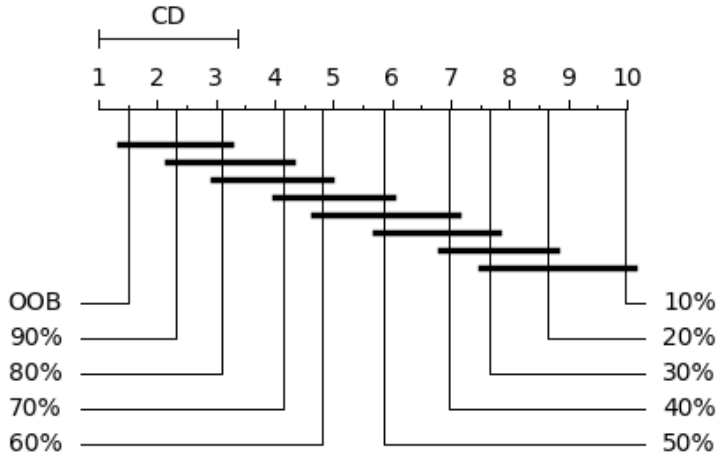

Figure 6: Nemenyi test of the results when estimating quality by kNN.

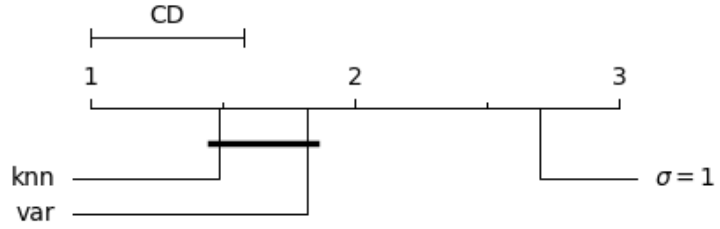

Figure 7: Nemenyi test of the results for all methods using oob.

4. Discussion

In this paper, the first, according to the best of our knowledge, large scale evaluation of approaches to generating split conformal predictive systems has been reported. We have mainly considered approaches that have earlier been demonstrated to be effective in the context of conformal regressors. One of the main results in our study shows that the oob approach is top-ranked more often than not, when compared to splitting the training data into a proper training and calibration set. This result is in line with previous results for conformal regressors, presented in (Johansson et al., 2014). Similar to when considering conformal regressors, one may explain the strong relative performance of the oob approach, by the stronger underlying models generated from the full training set, compared to using only a subset of the training examples. However, the difference between training using 90% versus 100% of the available data is marginal in many cases, which explains why a significant difference of the resulting conformal predictive systems was not observed in these cases. However, when comparing the oob method against using 70% or less of the data for training, there is indeed a significant difference in favor of the former. It should be noted that we have not investigated the case of using less than 10% for calibration when splitting the training set; results presented in (Vovk et al., 2019) showed that the performance started to decline when more than 90% was used for training, and we expect this to have been the case also in the considered setup. A major positive aspect of employing the oob approach is hence that there is one important parameter less to tune; the proper amount of data to use for training and calibration, respectively. The improvement in CRPS can be regarded as small in absolute values, but the relative improvement of several percent may have a large effect in decision making applications. Furthermore, no parameter tuning of the underlying model (random forests) was performed, which could have further improved the overall performance and perhaps also reduced the variance that was observed in the box plots. Tuning of the quality estimation methods parameters may also lead to that the performance is further improved and thus increasing the performance gap to not using quality estimation. However, we do not expect such tuning to dramatically change the outcome and conclusions of the study.

As discussed earlier, accurately estimating the quality of predictions can significantly improve the performance of SCPS. Similar to the results in earlier studies on conformal regression (Johansson et al., 2014; Boström et al., 2017), the methods used in this paper to estimate the quality of predictions together with SCPS were observed to significantly improve the performance. The results indicate that using kNN to estimate the quality appears to be the better option, which was also observed for conformal regressors in (Boström et al., 2017). However, the observed p-value of 0.366 do not allow for safely rejecting the null hypothesis, i.e., that there is no difference in performance between the kNN-based and the variance-based approach. Furthermore, since the computational cost of the kNN method is much higher than using the variance-based approach, the latter may be preferable.

5. Concluding remarks

We have investigated the effectiveness of alternative approaches to obtain calibration scores for conformal predictive systems; using various amounts of available training data for generating the underlying model and obtaining calibration scores, respectively, and using the

full training set for both model building and calibration, which is an option when bagging is employed for the former, as calibration scores can be calculated using out-of-bag predictions. Moreover, three approaches to estimate the quality of the individual predictions were considered; using a constant quality for all predictions, using the variance among the predictions of the individual trees of the underlying random forest and employing a k-nearest-neighbor method.

Results from a large empirical study were presented, using 33 publicly available datasets together with random forests as the underlying model, and using the continuous ranked probability score (CRPS) as a performance metric. The results show that compared to omitting the quality estimation of the individual predictions, a significant increase in performance is obtained by using either the variance or the k-nearest-neighbor method. As no significant difference between the two latter methods was observed, the use of the variance method could be motivated based on its lower computational cost. The results furthermore show that the use of out-of-bag examples for calibration is competitive with the most effective way of splitting the training data into a proper training set and a calibration set, without requiring tuning of the calibration set size.

One direction for future research concerns analyzing and evaluating the computational cost of the various approaches, something which has been mainly ignored in this study. Also, comparing the approaches in this study to the recently proposed cross-conformal predictive systems (Vovk et al., 2019) is a natural extension. The latter are computationally more costly than the considered approaches, but could potentially lead to improved performance. Another direction for future work includes investigating and evaluating additional approaches to estimating the quality of the individual prediction. An important direction for future work includes considering additional ways of evaluating the split conformal predictive systems, e.g., using alternative performance metrics that more directly relate to the use of such systems in decision-making contexts,

Acknowledgment

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation. HB was partly funded also by the Swedish Foundation for Strategic Research (CDA, grant no. BD15-0006).

References

Henrik Boström, Henrik Linusson, Tuve Löfström, and Ulf Johansson. Accelerating difficulty estimation for conformal regression forests. *Ann. Math. Artif. Intell.*, 81(1-2): 125–144, 2017.

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

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, 2006.

Dmitry Devetyarov and Ilia Nouretdinov. Prediction with confidence based on a random forest classifier. *Artificial Intelligence Applications and Innovations*, pages 37–44, 2010.

Gary Flake and Steve Lawrence. Efficient svm regression training with smo. *Machine Learning*, 46, 03 2001. doi: 10.1023/A:1012474916001.

Alexander Gammerman, Volodya Vovk, and Vladimir Vapnik. Learning by transduction. In *Proceedings of the Fourteenth conference on Uncertainty in Artificial Intelligence*, pages 148–155. Morgan Kaufmann, 1998.

Olga Ivina, Ilia Nouretdinov, and Alex Gammerman. Valid predictions with confidence estimation in an air pollution problem. *Progress in Artificial Intelligence*, 1:235–243, June 2012. ISSN 2192-6352. doi: 10.1007/s13748-012-0018-6.

Ulf Johansson, Henrik Boström, Tuve Löfström, and Henrik Linusson. Regression conformal prediction with random forests. *Machine Learning*, 97(1-2):155–176, 2014.

Harris Papadopoulos and Haris Haralambous. Neural networks regression inductive conformal predictor and its application to total electron content prediction. In *Artificial Neural Networks – ICANN 2010*, volume 6352 of *Lecture Notes in Computer Science*, pages 32–41. Springer Berlin Heidelberg, 2010.

Harris Papadopoulos and Haris Haralambous. Reliable prediction intervals with regression neural networks. *Neural Networks*, 24(8):842–851, 2011.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *Machine Learning: ECML 2002*, pages 345–356. Springer, 2002.

Harris Papadopoulos, Vladimir Vovk, and Alexander Gammerman. Regression conformal prediction with nearest neighbours. *Journal of Artificial Intelligence Research*, pages 815–840, 2011a.

Harris Papadopoulos, Vladimir Vovk, and Alexander Gammerman. Regression conformal prediction with nearest neighbours. *Journal of Artificial Intelligence Research*, 40(1): 815–840, 2011b.

Fredrik Svensson, Natalia Aniceto, Ulf Norinder, Isidro Cortes-Ciriano, Ola Spjuth, Lars Carlsson, and Andreas Bender. Conformal regression for quantitative structure–activity relationship modeling—quantifying prediction uncertainty. *Journal of Chemical Information and Modeling*, 58(5):1132–1140, 2018.

Vladimir Vovk and Ivan Petej. Venn–abers predictors. On-line Compression Modelling project (New Series), http://alrw.net, Working Paper 7, 2012.

Vladimir Vovk, Jieli Shen, Valery Manokhin, and Min-ge Xie. Nonparametric predictive distributions based on conformal prediction. On-line Compression Modelling project (New Series), http://alrw.net, Working Paper 17, 2017.

Vladimir Vovk, Ivan Petej, Ilia Nouretdinov, Valery Manokhin, and Alex Gammerman. Computationally efficient versions of conformal predictive distributions. *Neurocomputing*, 03 2019. doi: 10.1016/j.neucom.2019.10.110.

Appendix A.

Name	# instances	# attributes	source
abalone	4177	8	UCI
anacalt	4052	7	KEEL
bank8fh	8192	8	Delve
bank8fm	8192	8	Delve
bank8nh	8192	8	Delve
bank8nm	8192	8	Delve
boston	506	13	UCI
comp	8192	12	Delve
concreate	1030	8	UCI
cooling	768	8	UCI
deltaA	7129	5	KEEL
deltaE	9517	6	KEEL
friedm	1200	5	KEEL
heating	768	8	UCI
istanbul	536	7	UCI
kin8fh	8192	8	Delve
kin8fm	8192	8	Delve
kin8nh	8192	8	Delve
kin8nm	8192	8	Delve
laser	993	4	KEEL
mg	1385	6	(Flake and Lawrence, 2001)
mortage	1048	15	KEEL
plastic	1649	2	KEEL
puma8fh	8192	8	Delve
puma8fm	8192	8	Delve
puma8nh	8192	8	Delve
puma8nm	8192	8	Delve
quakes	2178	2	KEEL
stock	950	9	KEEL
treasury	1048	15	KEEL
wineRed	1599	11	UCI
wineWhite	4898	11	UCI
wizmir	1461	2	KEEL

Table A.1: Datasets used in the experiments.

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	OOB
abalone	0.044	0.042	0.041	0.042	0.041	0.040	0.040	0.041	0.040	0.040
anacalt	0.020	0.015	0.012	0.010	0.009	0.009	0.009	0.008	0.008	0.008
bank8fh	0.059	0.056	0.055	0.054	0.054	0.054	0.053	0.054	0.053	0.053
bank8fm	0.042	0.037	0.035	0.034	0.033	0.032	0.032	0.032	0.031	0.031
bank8nh	0.060	0.058	0.057	0.057	0.057	0.057	0.057	0.056	0.056	0.057
bank8nm	0.032	0.030	0.028	0.028	0.028	0.027	0.027	0.026	0.026	0.026
boston	0.066	0.051	0.047	0.045	0.044	0.042	0.043	0.039	0.040	0.039
comp	0.018	0.017	0.016	0.016	0.016	0.015	0.015	0.015	0.015	0.015
concreate	0.074	0.065	0.057	0.054	0.051	0.048	0.048	0.043	0.042	0.041
cooling	0.038	0.031	0.029	0.026	0.026	0.024	0.024	0.025	0.022	0.022
deltaA	0.021	0.021	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.020
deltaE	0.030	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029
friedm	0.059	0.055	0.052	0.048	0.046	0.047	0.046	0.045	0.045	0.043
heating	0.034	0.026	0.019	0.017	0.017	0.014	0.013	0.012	0.012	0.011
istanbul	0.047	0.045	0.045	0.044	0.047	0.044	0.044	0.043	0.045	0.043
kin8fh	0.048	0.046	0.045	0.044	0.044	0.043	0.043	0.043	0.043	0.043
kin8fm	0.036	0.032	0.031	0.030	0.029	0.028	0.027	0.027	0.026	0.026
kin8nh	0.080	0.077	0.077	0.075	0.075	0.074	0.073	0.073	0.073	0.073
kin8nm	0.075	0.071	0.068	0.066	0.065	0.064	0.063	0.063	0.062	0.061
laser	0.031	0.023	0.018	0.018	0.017	0.015	0.015	0.014	0.013	0.013
mg	0.068	0.060	0.054	0.052	0.049	0.047	0.047	0.046	0.043	0.044
mortage	0.013	0.009	0.007	0.006	0.005	0.005	0.005	0.005	0.004	0.004
plastic	0.101	0.097	0.098	0.100	0.097	0.098	0.101	0.099	0.098	0.097
puma8fh	0.087	0.085	0.084	0.084	0.083	0.083	0.083	0.083	0.082	0.082
puma8fm	0.058	0.052	0.049	0.047	0.046	0.045	0.044	0.043	0.043	0.043
puma8nh	0.094	0.089	0.088	0.086	0.085	0.084	0.083	0.083	0.083	0.082
puma8nm	0.079	0.070	0.066	0.063	0.060	0.059	0.057	0.057	0.055	0.055
quakes	0.099	0.097	0.099	0.099	0.098	0.098	0.098	0.097	0.097	0.097
stock	0.036	0.025	0.023	0.019	0.019	0.017	0.017	0.016	0.015	0.014
treasury	0.011	0.009	0.008	0.007	0.006	0.006	0.005	0.005	0.005	0.004
wineRed	0.074	0.072	0.070	0.069	0.068	0.066	0.066	0.064	0.064	0.063
wineWhite	0.067	0.064	0.063	0.062	0.060	0.059	0.058	0.056	0.056	0.055
wizmir	0.018	0.015	0.013	0.013	0.013	0.012	0.012	0.012	0.012	0.011

Table A.2: Mean CRPS for each dataset and training fraction size with $\hat{\sigma} = 1$.

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	OOB
abalone	0.043	0.041	0.040	0.041	0.041	0.039	0.039	0.040	0.039	0.039
anacalt	0.018	0.014	0.011	0.009	0.009	0.009	0.008	0.008	0.007	0.008
bank8fh	0.059	0.057	0.056	0.054	0.054	0.054	0.053	0.054	0.053	0.053
bank8fm	0.042	0.038	0.035	0.035	0.033	0.033	0.032	0.032	0.031	0.031
bank8nh	0.060	0.058	0.057	0.057	0.057	0.057	0.056	0.056	0.056	0.056
bank8nm	0.032	0.030	0.028	0.027	0.028	0.026	0.026	0.026	0.026	0.025
boston	0.065	0.051	0.046	0.044	0.043	0.041	0.041	0.038	0.039	0.038
comp	0.018	0.017	0.016	0.016	0.016	0.015	0.015	0.015	0.015	0.015
concreate	0.073	0.063	0.056	0.053	0.050	0.046	0.046	0.042	0.041	0.040
cooling	0.037	0.029	0.028	0.025	0.025	0.023	0.023	0.024	0.021	0.021
deltaA	0.021	0.020	0.020	0.020	0.020	0.020	0.020	0.019	0.020	0.019
deltaE	0.030	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029
friedm	0.060	0.056	0.052	0.048	0.047	0.047	0.046	0.046	0.045	0.044
heating	0.033	0.025	0.019	0.017	0.016	0.014	0.013	0.012	0.011	0.011
istanbul	0.047	0.045	0.045	0.044	0.047	0.044	0.044	0.043	0.045	0.043
kin8fh	0.048	0.046	0.045	0.044	0.044	0.043	0.043	0.043	0.042	0.042
kin8fm	0.036	0.033	0.031	0.029	0.029	0.028	0.027	0.027	0.026	0.026
kin8nh	0.079	0.077	0.076	0.074	0.074	0.074	0.073	0.073	0.072	0.072
kin8nm	0.075	0.070	0.067	0.066	0.064	0.064	0.062	0.062	0.061	0.060
laser	0.029	0.022	0.017	0.017	0.016	0.014	0.014	0.013	0.012	0.012
mg	0.063	0.056	0.050	0.047	0.044	0.042	0.042	0.041	0.038	0.039
mortage	0.012	0.009	0.007	0.006	0.005	0.005	0.005	0.005	0.004	0.004
plastic	0.106	0.100	0.100	0.101	0.098	0.100	0.102	0.100	0.100	0.098
puma8fh	0.087	0.084	0.083	0.083	0.082	0.082	0.082	0.082	0.081	0.081
puma8fm	0.059	0.053	0.049	0.048	0.046	0.045	0.044	0.044	0.043	0.043
puma8nh	0.093	0.089	0.087	0.085	0.084	0.083	0.082	0.082	0.082	0.082
puma8nm	0.078	0.070	0.066	0.063	0.060	0.059	0.057	0.057	0.055	0.055
quakes	0.100	0.099	0.100	0.100	0.098	0.099	0.100	0.098	0.098	0.098
stock	0.035	0.024	0.022	0.019	0.018	0.017	0.017	0.015	0.015	0.014
treasury	0.011	0.009	0.007	0.007	0.006	0.006	0.005	0.005	0.005	0.004
wineRed	0.074	0.072	0.069	0.068	0.067	0.065	0.064	0.062	0.062	0.062
wineWhite	0.067	0.064	0.062	0.061	0.058	0.058	0.057	0.055	0.054	0.053
wizmir	0.018	0.015	0.013	0.013	0.013	0.012	0.012	0.012	0.012	0.011

Table A.3: Mean CRPS for each dataset and training fraction size when the variance was used to determine $\hat{\sigma}$.

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	OOB
abalone	0.043	0.042	0.041	0.041	0.041	0.039	0.039	0.040	0.039	0.040
anacalt	0.019	0.015	0.011	0.009	0.009	0.009	0.008	0.008	0.007	0.007
bank8fh	0.059	0.056	0.055	0.054	0.054	0.054	0.053	0.054	0.053	0.053
bank8fm	0.042	0.037	0.035	0.034	0.033	0.032	0.031	0.032	0.030	0.030
bank8nh	0.060	0.058	0.057	0.057	0.057	0.057	0.056	0.056	0.056	0.056
bank8nm	0.031	0.029	0.028	0.027	0.027	0.026	0.026	0.025	0.026	0.025
boston	0.065	0.051	0.047	0.044	0.043	0.042	0.042	0.039	0.039	0.038
comp	0.018	0.017	0.016	0.015	0.015	0.015	0.015	0.015	0.015	0.014
concreate	0.074	0.064	0.057	0.053	0.050	0.047	0.047	0.043	0.041	0.040
cooling	0.037	0.029	0.028	0.024	0.024	0.022	0.022	0.023	0.020	0.020
deltaA	0.021	0.020	0.020	0.020	0.020	0.019	0.020	0.019	0.019	0.019
deltaE	0.030	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029	0.029
friedm	0.059	0.054	0.051	0.047	0.045	0.045	0.045	0.044	0.044	0.042
heating	0.034	0.026	0.019	0.017	0.016	0.014	0.013	0.012	0.011	0.011
istanbul	0.047	0.045	0.045	0.044	0.047	0.044	0.044	0.043	0.045	0.043
kin8fh	0.047	0.045	0.045	0.044	0.043	0.043	0.043	0.043	0.042	0.042
kin8fm	0.034	0.031	0.029	0.028	0.027	0.027	0.026	0.025	0.025	0.025
kin8nh	0.079	0.077	0.076	0.074	0.074	0.074	0.073	0.072	0.072	0.072
kin8nm	0.074	0.070	0.067	0.065	0.064	0.063	0.061	0.061	0.060	0.059
laser	0.030	0.022	0.017	0.017	0.016	0.014	0.014	0.013	0.012	0.013
mg	0.066	0.058	0.051	0.049	0.046	0.044	0.043	0.043	0.040	0.040
mortage	0.013	0.009	0.006	0.006	0.005	0.005	0.005	0.005	0.004	0.004
plastic	0.101	0.097	0.098	0.099	0.097	0.098	0.101	0.098	0.098	0.096
puma8fh	0.087	0.085	0.084	0.083	0.082	0.082	0.083	0.082	0.082	0.082
puma8fm	0.057	0.052	0.049	0.047	0.046	0.045	0.044	0.043	0.043	0.043
puma8nh	0.094	0.089	0.087	0.086	0.085	0.084	0.083	0.083	0.082	0.082
puma8nm	0.078	0.070	0.066	0.062	0.060	0.059	0.057	0.056	0.055	0.054
quakes	0.099	0.098	0.099	0.099	0.097	0.098	0.098	0.097	0.097	0.097
stock	0.036	0.024	0.022	0.019	0.019	0.017	0.017	0.015	0.015	0.014
treasury	0.011	0.009	0.007	0.006	0.006	0.006	0.005	0.005	0.005	0.004
wineRed	0.074	0.072	0.070	0.068	0.068	0.066	0.066	0.064	0.064	0.063
wineWhite	0.067	0.064	0.063	0.062	0.059	0.059	0.058	0.056	0.056	0.054
wizmir	0.018	0.015	0.013	0.013	0.012	0.012	0.012	0.012	0.012	0.011

Table A.4: Mean CRPS for each dataset and training fraction size when the kNN method was used to determine $\hat{\sigma}$.