Proceedings of Machine Learning Research 152:1–20, 2021 Conformal and Probabilistic Prediction and Applications

Impact of Model-Agnostic Nonconformity Functions on Efficiency of Conformal Classifiers: an Extensive Study

Marharyta Aleksandrova MARHARYTA.ALEKSANDROVA@{UNI.LU,GMAIL.COM}
University of Luxembourg, 2 avenue de l'Université, L-4365 Esch-sur-Alzette, Luxembourg

Oleg Chertov CHERTOV@I.UA
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Applied Mathematics Department, 14-A Politekhnichna St, 03056 Kyiv, Ukraine

Editor: Lars Carlsson, Zhiyuan Luo, Giovanni Cherubin and Khuong An Nguyen

Abstract

The property of conformal predictors to guarantee the required accuracy rate makes this framework attractive in various practical applications. However, this property is achieved at a price of reduction in precision. In the case of conformal classification, the system can output multiple class labels instead of one. It is also known, that the choice of nonconformity function has a major impact on the efficiency of conformal classifiers. Recently, it was shown that different model-agnostic nonconformity functions result in conformal classifiers with different characteristics. For a Neural Network-based conformal classifier, the *inverse probability* (or hinge loss) allows minimizing the average number of predicted labels, and *margin* results in a larger fraction of singleton predictions. In this work, we aim to further extend this study. We perform an experimental evaluation using 8 different classification algorithms and discuss when the previously observed relationship holds or not. Additionally, we propose a successful method to combine the properties of these two nonconformity functions.

Keywords: Conformal classification, Nonconformity functions, Efficiency

1. Introduction

Conformal prediction (Shafer and Vovk, 2008; Vovk et al., 2005) is a framework that produces predictions with accuracy guarantees. For a given value of significance level $\epsilon \in (0, 1)$, a conformal predictor is guaranteed to make exactly ϵ errors in the long run. This is achieved at a price of a reduction in prediction precision. Instead of predicting a single class label, in the case of classification, or a single number, in the case of regression, a conformal predictor outputs a range prediction, that is a set of class labels or an interval that contains the true value with probability $1 - \epsilon$. Construction of a conformal predictor with $\epsilon = 0$ is a trivial task. It is enough to output all class labels or an unbounded interval in case of classification and regression respectively. However, such a predictor is of low value, that is, it is not *efficient*. The question thus is how to guarantee the given level of error rate (ϵ) by producing the smallest prediction regions. This property is achieved via the definition of a proper nonconformity function that succeeds to measure the *strangeness* or *nonconformity* of every data instance (Shafer and Vovk, 2008).

In the case of classification, the *efficiency* of a conformal predictor is often measured in terms of 2 metrics: *avgC*, which stands for the average number of predicted class labels per instance, and *oneC*, which stands for the fraction of produced singleton predictions.

Naturally, one would want to minimize *avgC* and maximize *oneC* at the same time. A recent study by Johansson et al. (2017) showed that *the usage of margin as a nonconformity function results in higher oneC and the usage of inverse probability (also known as hinge) results in lower values of avgC*[1]. The authors used 21 datasets to demonstrate the statistical significance of this relationship. However, this was done for the case where the baseline classifiers were either a single artificial neural network (ANN) or an ensemble of bagged ANNs. In this paper, we aim to extend this study with the following contributions:

1. We study if the same pattern is present when other classification algorithms are used. Our experimental results with 8 different classifiers and 9 publicly available datasets show that although the previously observed pattern does hold in the majority of the cases, the choice of the best nonconformity function can depend on the analyzed dataset and the chosen underlying classification model. For example, k-nearest neighbours classifier performs best with *margin*. *Margin* is also the best choice in case of `balance` dataset regardless of the chosen classification model.

2. We propose a method to combine both nonconformity functions. Our experimental evaluation shows that this combination always results in better or the same performance as *inverse probability*, thus allowing to increase the value of *oneC* and decrease the value of *avgC*. In some cases, the proposed combination outperforms both *inverse probability* and *margin* in terms of both efficiency metrics.

3. We discuss several aspects of how the accuracy of the baseline classifier can impact the performance of the resulting conformal predictor. In particular, if the baseline prediction accuracy is very good, then nonconformity function has no impact on the efficiency. Also, the accuracy of the baseline classifier strongly correlates with the fraction of singleton predictions that contain the true label. In this way, the accuracy can be an indicator of the usefulness of the *oneC* metric.

The rest of the paper is organized as follows. In Section 2 we discuss related work. Section 3 is dedicated to the description of the proposed strategy to combine advantages of *margin* and *inverse probability* nonconformity functions. Section 4 and Section 5 present the experimental setup and results. Finally, we summarize our work in Section 6.

2. Related work

Conformal prediction is a relatively new paradigm developed at the beginning of 2000, see Linusson (2021) for an overview. It was originally developed for transductive setting (Vovk, 2013). The latter is efficient in terms of data usage but is also computationally expensive. Recent studies, including the current one, focus on *Inductive Conformal Prediction* (*ICP*) (Papadopoulos, 2008). *ICP* trains the learning model only once, however a part of the training dataset should be put aside for model calibration using a predefined nonconformity function.

There are two groups of nonconformity functions: *model-agnostic* and *model-dependent*. Model-dependent nonconformity functions are defined based on the underlying prediction model. Such functions can depend on the distance to the separating hyperplane in

1. In the rest of the text, we will refer to this relationship as a baseline or original pattern (relationship).

SVM (Balasubramanian et al., 2009), or the distance between instances in KNN classifier (Proedrou et al., 2002). These nonconformity functions are model-specific, thereby, one can not draw generalized conclusions about their behaviour. In a recent study by Johansson et al. (2017) it was shown that model-agnostic nonconformity functions do have some general characteristics. *Inverse probability* nonconformity function, also known as *hinge*, is defined by the equation $\Delta\left[h(\boldsymbol{x_i}), y_i\right] = 1 - \hat{P}_h(y_i|\boldsymbol{x_i})$, where $\boldsymbol{x_i}$ is the analyzed data instance, y_i is a tentative class label, and $\hat{P}_h(y_i|\boldsymbol{x_i})$ is the probability assigned to this label given the instance $\boldsymbol{x_i}$ by the underlying classifier h. It was shown that conformal classifiers based on this metric tend to generate prediction regions of lower average length ($avgC$). At the same time, *margin* nonconformity function results in a larger fraction of singleton predictions ($oneC$). The latter is defined by the following formula $\Delta\left[h(\boldsymbol{x_i}), y_i\right] = \max_{y \neq y_i} \hat{P}_h(y|\boldsymbol{x_i}) - \hat{P}_h(y_i|\boldsymbol{x_i})$, and it measures how different is the probability of the label y_i from the probability of another most probable class label. The experimental evaluations in Johansson et al. (2017), however, were performed for a limited number of underlying classification models: ANN and ensemble of bagged ANNs. To the best of our knowledge, there are no research works dedicated to the validity analysis of the discovered pattern in the case of other classification algorithms. This piece of research is missing to draw global conclusions about the characteristics of these nonconformity functions.

Combining characteristics of both *margin* and *inverse probability* nonconformity functions is a tempting idea. In recent years many authors dedicated their efforts to understand how one can generate more efficient conformal predictions through a combination of several conformal predictors. Yang and Kuchibhotla (2021) and Toccaceli and Gammerman (2019) studied how to aggregate conformal predictions based on different training algorithms. Various strategies were proposed for such combination: via p-values (Toccaceli and Gammerman, 2017), a combination of monotonic conformity scores (Gauraha and Spjuth, 2018), majority voting (Cherubin, 2019), out-of-bag calibration (Linusson et al., 2020), or via established results in classical statistical hypothesis testing (Toccaceli, 2019). The challenge of every combination of conformal predictors is to retain *validity*, that is to achieve the empirical error rate not exceeding the predefined value ϵ. This property is usually demonstrated experimentally and some authors provide guidelines on which values of significance levels should be used for individual conformal algorithms to achieve the desired validity of the resulting combination. As opposed to these general approaches, in Section 3 we propose a procedure that is based on the properties of *margin* and *inverse probability*. Through experiments, we show that this approach allows combining their characteristics, higher $oneC$ and lower $avgC$, and retains the validity at the same time.

3. Combination of *inverse probability* and *margin* nonconformity functions

As was shown by Johansson et al. (2017), the usage of *inverse probability* nonconformity function results in less number of predicted class labels on average (lower $avgC$), and *margin* results in a larger fraction of singleton predictions (higher $oneC$). In this section, we propose an approach to combine these properties of the two nonconformity functions. The validity of this method is studied empirically in Section 5.1 and its efficiency is demonstrated in Section 5.3.

It is desirable to have more singleton predictions. However, if a singleton prediction does not contain the true label, then the metric *oneC* not only loses its value but also becomes misleading. In Section 5.2 we demonstrate that for some datasets only a half of singleton predictions contain the true label. Hence, in our proposed method we decide to take the results produced by *inverse probability* nonconformity function as a baseline, and then extend them with some singleton predictions resulting from the usage of *margin*.

The proposed procedure is the following. **First**, we construct conformal predictors using both nonconformity functions separately[2]. For the conformal predictor based on *inverse probability*, we use the value of ϵ specified by the user as the significance level. For the conformal predictor based on *margin*, we set the significance level equal to $\epsilon/2$. This is done to compensate for possible erroneous singleton predictions produced by *margin* nonconformity function and to achieve the required level or empirical error rate. **Second**, for every instance in the testing or production dataset, we analyze the predictions generated by both conformal classifiers. If the conformal classifier based on *margin* outputs a singleton and the other conformal classifier does not, then the prediction is taken from the first model. Otherwise, the output of the conformal classifier based on *inverse probability* is used. Such a combination will perform in the worst case the same as the conformal predictor based on *inverse probability*. Otherwise, the values of *oneC* and/or *avgC* will be improved, as some non-singleton predictions will be replaced with singletons. Thereby, in case the validity is preserved, this combination can be considered as an improved version of the *inverse probability* nonconformity function.

In the rest of the paper, we use *M* and *IP* to refer to *margin* and *inverse probability* respectively. *IP_M* is used to refer to the combination explained above. For simplicity, sometimes *IP_M* is referred to as a nonconformity function, although technically it is not.

4. Experimental setup

To perform experimental analysis, we used the implementation of conformal prediction framework available from *nonconformist*[3] Python library. We followed the general experimental setup from the original paper by Johansson et al. (2017). That is we used 10x10-fold cross-validation with 90% of the data used for training and validation of the model, and 10% used for testing. The training dataset was further split into a proper training set and a calibration set in proportion 4:1. All the results reported below are averaged over the 10x10 folds. In the original study, the authors used 21 publicly available multi-class datasets from UCI repository Dua and Graff (2017). In this paper, we present not aggregated, but detailed results for every analyzed dataset. Such a setup allows us to identify dataset-specific relationships. That is why we chose 9 representative datasets with different characteristics from the original list of 21 ones. The general information about these datasets, such as the number of instances, attributes, and defined classes is given in the first section of Table 1.

The original study by Johansson et al. (2017) analyzed the performance of conformal classifiers based on the ANN classification model. In this paper, we aim to further extend this analysis and use 8 different classification algorithms as baseline models: Support Vec-

2. See Vovk et al. (2005); Shafer and Vovk (2008); Johansson et al. (2017) for explanation of how conformal predictors are constructed.
3. https://github.com/donlnz/nonconformist

Table 1: Characteristics of the datasets and summarization of results

Section		**Datasets**	**iris**	*user*	**glass**	*cars*	**wave**	**balance**	*wineR*	*wineW*	*yeast*
1	info	# instances	150	403	214	1728	5000	625	1599	4898	1484
		# attributes	4	5	9	6	21	4	11	11	8
		# classes	3	4	6	4	3	3	6	7	10
		balanced	Y	mostly	N	N	Y	N	N	N	N
2	*b_err* range, %		2-6	5-32	24-92	7-96	13-25	3-21	38-50	45-58	40-87
	b_err median, %		4	7	48	13	17	10	45	53	44
3	*E_oneC*	mean	0.98	0.93	0.63	0.88	0.90	0.96	0.52	0.50	0.56
		mean-std	0.00	0.00	0.02	0.00	0.00	0.00	0.03	0.02	0.02
		corr. *b_acc*	0.92	0.98	0.69	0.93	0.92	0.80	0.27	0.78	0.97
4	stat. thres.	*oneC*,%	5	15	75	47.5	25	67.5	72.5	65	77.5
		avgC,%	2.5	22.5	67.5	37.5	27.5	42.5	80	77.5	82.5
	stat.	*oneC*,%		5	50	45	30	55	82.5	85	85
		avgC,%		17.5	37.5	45	47.5	50	82.5	77.5	77.5
5	patterns	*M* is the best		**KNN**	**KNN**, DT	**KNN**	Ada (0.1, 0.15, 0.20)	**KNN**, DT, RF, GNP, MPR, Ada, QDA	**KNN** (0.05, 0.1), DT(0.05)	**KNN** (0.05, 0.1)	**KNN**(0.05, 0.1, 0.15), DT(0.05), Ada(0.15, 0.2), QDA(0.15, 0.2)
		IP is the best		Ada(0.15)			Ada(0.01)				
		IP_M is the best			MPR	RF	RF				
		IP_M > *IP*	Y	Y	Y	Y	Y	Y	Y	Y	Y

tor Machine (SVM), Decision Tree (DT), k-Nearest Neighbours (KNN), AdaBoost (Ada), Gaussian Naive Bayes (GNB), Multilayer Perceptron (MPR), Random Forest (RF) and Quadratic Discriminant Analysis (QDA). We used implementations of these algorithms available from *scikit-learn* Python library. In Table 2, we summarize the input parameters of these algorithms unless the default values are used.

Table 2: Input parameters of classification algorithms

Algorithm	Input parameters
SVM	probability=True
DT	min_samples_split=max(5, 5% of proper training dataset)
KNN	n_neighbors=5
MPR	alpha=1, max_iter=1000
RF	n_estimators=10, min_samples_split=0

Different classifiers perform differently on different datasets. We demonstrate this in the second section of Table 1. To calculate the corresponding values, we used the same 10x10-fold cross-validation but without splitting the training set into a proper training set and a validation set. *b_err range* from the first row of this section demonstrates the range of errors produced by all 8 classification algorithms in the baseline mode[4]. We can notice that some datasets are easier to classify, for example, `iris` dataset for which the maximum error is 6%. At the same time, other datasets are more difficult, for example, `wineW` for which none of the classifiers can produce error less than 45%. The performance of classifiers is not uniformly distributed within the given ranges. This can be seen from the median of baseline error distribution, see row *b_err median*. For example, for the `cars` dataset different classifiers result in errors ranging from 7% to 96%. However, the median value of 13 shows that half of them perform relatively well.

All experimental evaluations were performed for 5 different values of significance level $\epsilon \in \{0.01, 0.05, 0.1, 0.15, 0.20\}$. For every combination of dataset, baseline classification algorithm and ϵ, we calculated the values of *oneC* and *avgC* with 2 different nonconformity functions (*IP* and *M*) and their combination *IP_M*. After that, the results were compared to see if any of the nonconformity functions or their combination results in a more efficient conformal predictor. Due to the space limitations of this paper, we present detailed results only for 4 datasets highlighted in bold in Table 1. However, experimental results for all datasets and the code used for the experimentation are available in the related git repository[5].

5. Experimental results

5.1. Validity

We start with the analysis of *validity*, that is first we check if the produced conformal predictors indeed achieve the required error rate. This property was demonstrated in previous

4. In this text we use the *baseline mode* to refer to the standard (non-conformal) prediction.
5. https://github.com/marharyta-aleksandrova/copa-2021-conformal-learning

works both for *inverse probability* and *margin*. It is also theoretically guaranteed for any nonconformity function, but not for a combination of those, like *IP_M*. In Table 3 we demonstrate the empirical error rates averaged among all datasets. As we can see, all conformal predictors are well-calibrated. The validity of conformal predictor based on *IP_M* can be explained by the fact, that we add *margin*-based predictions to the *IP*-based model only in case when we are very confident about them. Recall that the significance level is set to $\epsilon/2$ for this case, see Section 3. Thereby, the probability to generate enough invalid predictions to surpass the allowed error rate ϵ is very low.

Table 3: Empirical error rates

eps:	**0.01**	**0.05**	**0.10**	**0.15**	**0.20**
IP	0.01	0.05	0.09	0.14	0.19
IP_M	0.01	0.05	0.10	0.15	0.19
M	0.01	0.05	0.10	0.15	0.19

5.2. Informativeness of *oneC*

In Section 3, we discussed the issue that can happen with *oneC* metric. Indeed, if a large portion of predicted singletons does not contain the true label, then this metric can be misleading. We calculated the ratio of the number of singleton predictions that contain the true label to the overall number of singleton predictors for different setups and algorithms. We denote this value as *E_oneC* from *effective oneC*. The corresponding results are presented in section 3 of Table 1.

The first row of this section shows the averaged value of *E_oneC* over all 5 values of ϵ and 3 nonconformity functions. We can notice that this value is very different for different datasets ranging from 0.98 for `iris` to only 0.50 for `wineW`. This means that for `wineW` on average half of the produced singleton predictions do not contain the true label. In real applications, this prediction can be more confusing than a prediction with multiple labels.

We can notice that there is a certain correlation between the mean value of *E_oneC* and the difficulty of the dataset for the baseline classifiers (*b_err*). To analyze this relationship, we calculated the value of correlation between the corresponding characteristics. The results are presented in the third row *corr. b_acc*. We can see that for 5 of 9 datasets (56%) the correlation is above 0.9. This holds for `iris`, `user`, `cars`, `wave`, and `yeast` datasets. For 2 more datasets (`balance` and `wineW`), the correlation coefficient is approximately 0.8. For `glass` dataset, it is equal to 0.69, and only for `wineR` the correlation is as low as 0.27. These results show a strong relationship between the baseline error of the underlying classification model and the correctness of singleton predictions.

Finally, to check if *E_oneC* depends on the chosen nonconformity function, we averaged the results separately for different non-conformity functions and then calculated the standard deviation of the resulting three values. The corresponding results are presented in the second row *mean-std*. We notice that *mean-std* is very low for all datasets. This indicates that *E_oneC* does not depend on the choice of nonconformity function.

5.3. Efficiency of different nonconformity functions

In this section, we study the relationship between different nonconformity functions and the effectiveness of the resulting conformal predictors. For every combination of a dataset, a baseline classifier, and a value of ϵ, we calculate the values of *oneC* and *avgC*. For visual analysis, the corresponding results are plotted using the same format as in Figs. 1 and 2. Such figures contain information about the distribution of instances between classes (plots *a*), the baseline error rate of all classification algorithms *b_err* (plots *b*), and the corresponding values of the efficiency metrics (plots from *c* to *j*). The latter group of plots contains three lines corresponding to *margin* (dashed line), *inverse probability* (dash and dot line) and their combination *IP_M* (thin solid line).

Further, we evaluate how significant are the differences between different nonconformity functions. The corresponding results are summarized in the same format as presented in Table 4. Here, for every baseline classifier and value of ϵ, we present a comparison matrix. A value in the matrix shows if the row setup is better (indicated with +) or worse (indicated with −) than the column setup. The star indicates if the detected difference is statistically significant[6]. To avoid too small differences, we put a sign into the matrix only if the corresponding difference is above the threshold of 2%[7] or it is statistically significant. For example, from Table 4 we can see that *margin* results in better values of *oneC* than *IP* and *IP_M* for SVM with $\epsilon = 0.05$. These results are also statistically significant, as indicated by a *. At the same time, for $\epsilon = 0.1$ *margin* improves the results of *IP_M* by at least 2%. However, this difference is not statistically significant. Section 4 of Table 1 shows the fraction of setups, for which we can observe a difference between the performance of conformal classifiers with different nonconformity functions either by exceeding the threshold of 2% (*thres.*) or observing statistical significance (*stat.*). These values are calculated as follows. For every dataset, we have 40 setups (5 values of ϵ x 8 baseline classifiers). Each such setup corresponds to one matrix for *oneC* and one matrix for *avgC* in significance tables. We calculate how many of these matrices either have at least one + or −, or have at least one statistically significant result. After that, the calculated number is divided over 40.

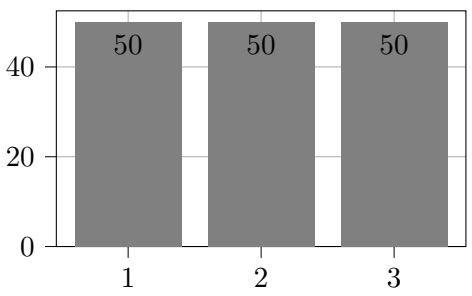
Using such information for all datasets, we can analyze the efficiency of conformal classifiers for different nonconformity functions and identify which of them perform better. The corresponding findings are summarized in section 5 of Table 1. This part of the table shows the deviations from the pattern originally observed by Johansson et al. (2017). In our experiments we observed the following 3 deviations: 1) ***M* is the best**: *margin* can produce both higher values of *oneC* and lower values of *avgC*, that is *margin* is the best choice of nonconformity function; 2) ***IP* is the best**: *inverse probability* is the best choice of nonconformity function; 3) ***IP_M* is the best**: the combination of *M* and *IP* produces the best results in terms of both efficiency metrics. Additionally, our experiments show that *IP_M* never performs worse than *IP* (*IP_M* > *IP*). Note also, that we never observed the inverse pattern, that is *inverse probability* resulting in higher values of *oneC* and *margin* resulting in lower values of *avgC* at the same time. In the rest of this section, we demonstrate our

6. Statistical significance was estimated using Student's t-test with $\alpha = 0.05$.
7. For 100% we take the value of 1 for *oneC* and the total number of classes for *avgC*. These are the maximum values of these two metrics.

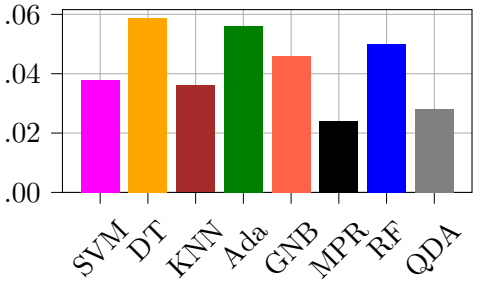
main findings on the examples of 4 datasets highlighted in bold in Table 1. The general conclusions are discussed in Section 5.4.

IRIS. The results for `iris` dataset are presented in Fig. 1. As we can see from the plots comparing *oneC* and *avgC*, there is almost no difference and all 3 nonconformity functions produce the same results. This is also reflected in section 4 of Table 1. Only for 5% of setups we can observe a difference in terms of *oneC*, see performance of RF in Fig. 1(*i*). The difference in *avgC* is observed even less often, only in 2.5% of setups, see results for GNP in Fig. 1(*h*). Statistically significant differences are never observed. This means that the table with the significance of results like Table 4 for this dataset is almost empty. That is why we do not present it in the paper. As expected, not many patterns can be observed in this case. The only pattern that we observe, is *IP_M* > *IP*. Note, that this dataset is perfectly balanced, see Fig. 1(*a*) and all classifiers have very good performance with maximum error not exceeding 6%, see Table 1 and Fig. 1(*b*). We can also notice that there is a relationship between the baseline error of the classifier and the efficiency of the resulting conformal predictor: better classifiers tend to produce conformal predictors of better quality. The 3 worst baseline predictors DT, Ada, and RF also produce conformal predictors with lower values of *oneC*, see Figs. 1(*c*), 1(*e*) and 1(*i*) and larger values of *avgC*, see Figs. 1(*d*) and 1(*j*).

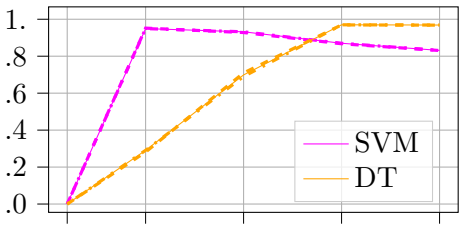
GLASS. Next, we analyze the results for `glass` dataset presented in Fig. 2 and Table 4. This dataset is unbalanced, see Fig. 2(*a*) and different classifiers have different performance with baseline error ranging from 24% for RF to 92% for QDA, see Fig. 2(*b*). As it was observed for `iris` dataset, those classifiers that perform better in the baseline scenario also tend to produce more efficient conformal predictors. For example, see the results for RF in Figs. 2(*i*) and 2(*j*), and the results for KNN in Figs. 2(*e*) and 2(*f*). At the same time, classifiers that perform badly in the baseline scenario produce conformal predictors of low quality, see results for QDA in Figs. 2(*i*) and 2(*j*). There are also exceptions, but they are less numerous. For example, for this dataset the baseline performance of the DT classifier is good (*b_err* $\approx$ 30%), however, the resulting conformal classifier is less efficient than the one based on SVM with *b_err* > 60%. For this dataset, we can also observe a clear difference in performance depending on which nonconformity function is used. This is summarized in Table 4. Analyzing the results presented in this table, we can observe the following patterns. 1) For KNN and DT-based conformal predictors *margin* is the best choice of nonconformity function. Indeed, this is reflected in Figs. 2(*c*) and 2(*e*) (*margin* results in higher values of *oneC*) and Figs. 2(*d*) and 2(*f*) (*margin* also results in lower values of *avgC*). This pattern is also shown in Table 4 in comparison matrices corresponding to KNN and DT algorithms. We can see that in all comparison matrices, there is a + in rows corresponding to *margin* indicating that it outperforms both *IP* and *IP_M*, except *avgC* for KNN with $\epsilon = 0.2$. 2) For MPR *IP_M* is the best choice of nonconformity function. This is reflected in the corresponding comparison matrices of Table 4, from which we can see that *IP_M* results in higher values of *oneC* and lower values of *avgC* at the same time. This is also visible in Figs. 2(*g*) and 2(*h*). 3) Finally, we never observe *IP_M* being outperformed by *IP*. In the inverse direction, however, *IP_M* does improve the results of *IP*. For example, for SVM with $\epsilon = 0.05$ or $\epsilon = 0.1$ *IP_M* allows to achieve better values of *oneC* and this improvement is also statistically significant.



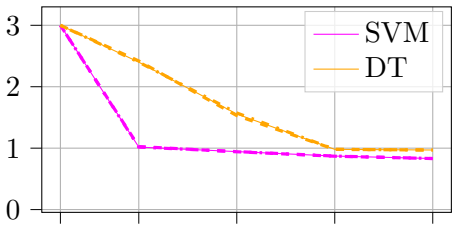
(*a*) Distribution between classes



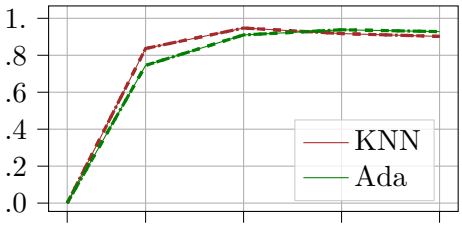
(*b*) Baseline error, *b_err*



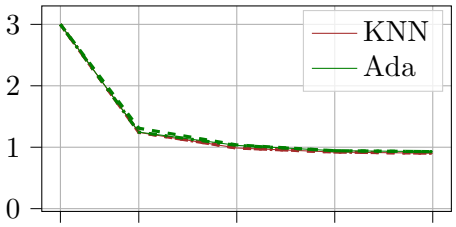
(*c*) *oneC*: SVM and DT



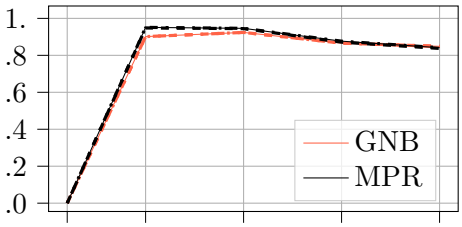
(*d*) *avgC*: SVM and DT



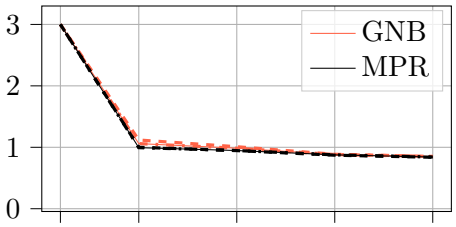
(*e*) *oneC*: KNN and Ada



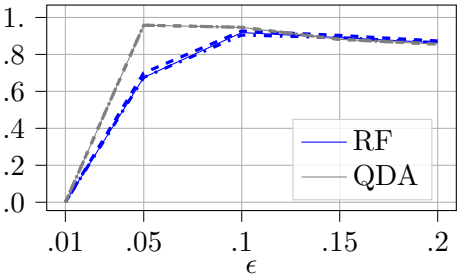
(*f*) *avgC*: KNN and Ada



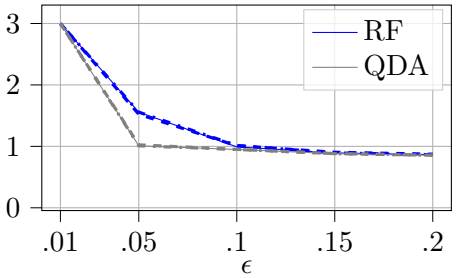
(*g*) *oneC*: GNB and MPR



(*h*) *avgC*: GNB and MPR

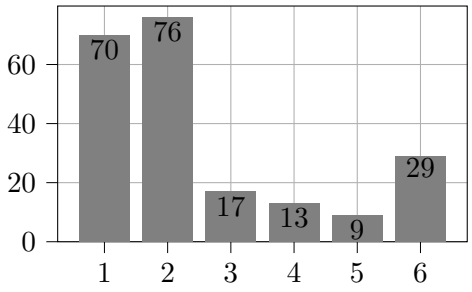


(*i*) *oneC*: RF and QDA

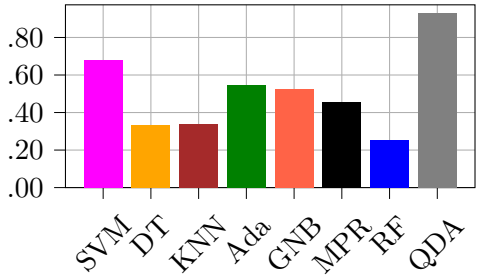


(*j*) *avgC*: RF and QDA

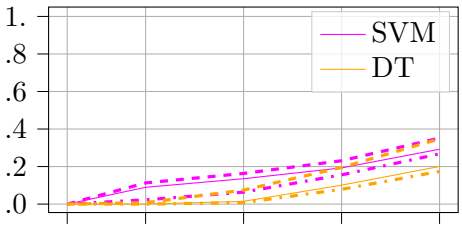
Figure 1: Results for *iris*: *M* - dashed line, *IP* - dash and dot line, *IP_M* - thin solid line



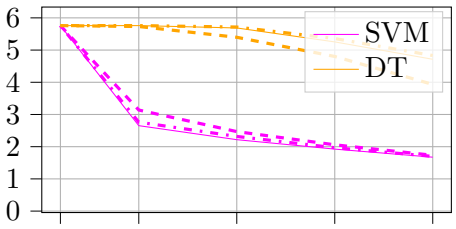
(*a*) Distribution between classes



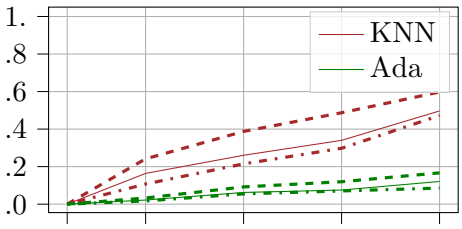
(*b*) Baseline error, *b_err*



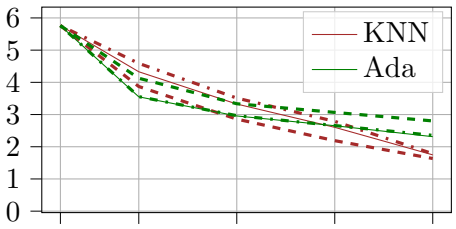
(*c*) *oneC*: SVM and DT



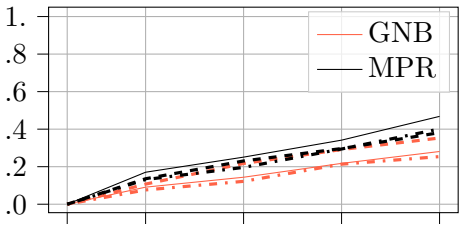
(*d*) *avgC*: SVM and DT



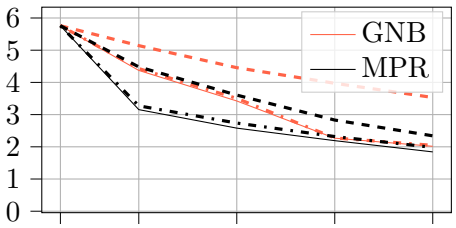
(*e*) *oneC*: KNN and Ada



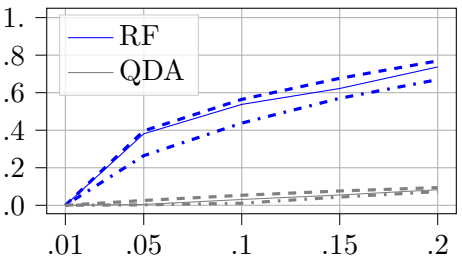
(*f*) *avgC*: KNN and Ada



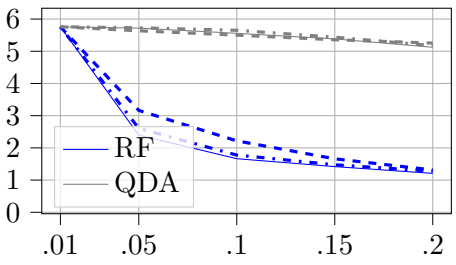
(*g*) *oneC*: GNB and MPR



(*h*) *avgC*: GNB and MPR



(*i*) *oneC*: RF and QDA



(*j*) *avgC*: RF and QDA

Figure 2: Results for *glass*: *M* - dashed line, *IP* - dash and dot line, *IP_M* - thin solid line

Table 4: Significance of results for the *glass* dataset

		$\epsilon = 0.01$			$\epsilon = 0.05$			$\epsilon = 0.1$			$\epsilon = 0.15$			$\epsilon = 0.2$		
oneC	ip					$-^*$	$-^*$		$-^*$	$-^*$		$-$	$-^*$		$-$	$-^*$
	ip_m				$+^*$		$-^*$	$+^*$		$-$	$+$		$-$	$+$		$-$
	m				$+^*$	$+^*$		$+^*$	$+$		$+^*$	$+$		$+^*$	$+$	
SVM		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip						$+^*$			$+$						
	ip_m						$+^*$			$+^*$			$+$			
	m				$-^*$	$-^*$		$-$	$-^*$			$-$				
oneC	ip									$-$		$-$	$-$		$-$	$-$
	ip_m									$-$	$+$		$-$	$+$		$-$
	m							$+$	$+$		$+$	$+$		$+$	$+$	
DT		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip									$-$			$-$			$-$
	ip_m									$-$			$-$			$-$
	m							$+$	$+$		$+$	$+$		$+$	$+$	
oneC	ip					$-$	$-^*$		$-$	$-^*$		$-$	$-^*$		$-$	$-^*$
	ip_m				$+$		$-$	$+$		$-$	$+$		$-^*$	$+$		$-$
	m				$+^*$	$+$		$+^*$	$+$		$+^*$	$+^*$		$+^*$	$+$	
KNN		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip					$-$	$-$		$-$	$-$		$-$	$-$			$-$
	ip_m				$+$		$-$	$+$		$-$	$+$		$-$			
	m				$+$	$+$		$+$	$+$		$+$	$+$		$+$		
oneC	ip									$-^*$			$-^*$		$-^*$	$-^*$
	ip_m									$-$			$-^*$	$+^*$		$-^*$
	m							$+^*$	$+$		$+^*$	$+^*$		$+^*$	$+^*$	
Ada		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip						$+^*$			$+^*$			$+^*$			$+^*$
	ip_m						$+^*$			$+^*$			$+^*$			$+^*$
	m				$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$	
oneC	ip						$-$		$-$	$-^*$			$-^*$		$-$	$-^*$
	ip_m							$+$		$-^*$			$-^*$	$+^*$		$-^*$
	m				$+$			$+^*$	$+^*$		$+^*$	$+^*$		$+^*$	$+^*$	
GNB		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip						$+$			$+^*$			$+^*$			$+^*$
	ip_m						$+$			$+^*$			$+^*$			$+^*$
	m				$-$	$-$		$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$	
oneC	ip					$-$			$-$	$-$		$-$			$-$	
	ip_m				$+$		$+$	$+$		$+$	$+$		$+$	$+$		$+$
	m					$-$		$+$	$-$			$-$			$-$	
MPR		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip						$+^*$		$-$	$+^*$		$-$	$+^*$		$-$	$+$
	ip_m						$+^*$	$+$		$+^*$	$+$		$+^*$	$+$		$+^*$
	m				$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$		$-$	$-^*$	
oneC	ip					$-$	$-^*$		$-$	$-^*$		$-$	$-^*$		$-$	$-^*$
	ip_m				$+$			$+$		$-$	$+$		$-$	$+$		$-$
	m				$+^*$			$+^*$	$+$		$+^*$	$+$		$+^*$	$+$	
RF		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip					$-$	$+$			$+^*$			$+$			
	ip_m				$+$		$+^*$			$+^*$			$+$			
	m				$-$	$-^*$		$-^*$	$-^*$		$-$	$-$				
oneC	ip						$-^*$			$-^*$			$-$			$-$
	ip_m						$-^*$			$-$			$-$			
	m				$+^*$	$+^*$		$+^*$	$+$		$+$	$+$		$+$		
QDA		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip									$-$						
	ip_m															$+$
	m							$+$							$-$	

WAVE. The results for `wave` dataset are presented in Fig. 3 and Table 5. This dataset is balanced, see Fig. 3(*b*), and overall performance of classifies is quite good, see Fig. 3(*b*). There is only one classifier, DT, wich results in $b\_err > 20\%$. As it was also noted for `iris`, when classifiers have good baseline performance, the difference between different nonconformity functions diminishes. This is reflected in the reduction of values in section 4 of Table 1 (25%, 27.5%, 30%, and 47.5%) as compared to the corresponding values for `glass` dataset (75%, 67.5%, 50%, and 37.5%). We can observe visual differences in Fig. 3 only for DT, Ada, GNB, and RF. No difference is observed for KNN, however, despite it having a comparable value of b_err. For this dataset, we can also see that *margin* results in best performance for $\epsilon \in \{0.1, 0.15, 0.2\}$ when Ada classifier is used. This improvement is also statistically significant, see Table 5. In the case of $\epsilon = 0.01$ however, the best performance is achieved by *IP* nonconformity function, and this result is also statistically significant. Further, *IP_M* is the best choice for RF classifier with statistical significance of the improvement in most of the settings.

BALANCE. The case of `balance` dataset is very interesting: for most of the classification models *margin* is always the best choice of nonconformity function. As reflected in Table 6 and supported by Fig. 4, these differences are mostly statistically significant. The only cases for which the original pattern holds are SVM with all values of ϵ and QDA with $\epsilon = 0.05$. For this dataset, we also observe differences between different nonconformity functions despite the relatively good performance of many classifiers.

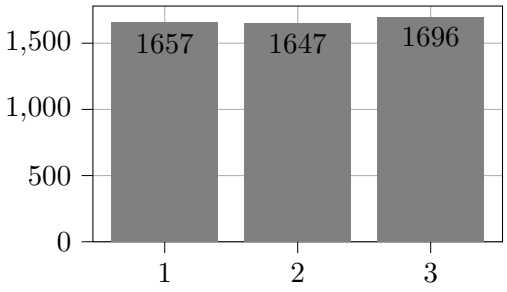
5.4. Summary of results

In this subsection we summarize the findings from our experimental results shown in Table 1.

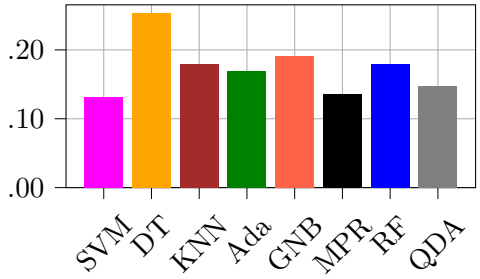
As we saw, ***margin* can be the best choice of nonconformity function** for some datasets (`balance` dataset) or some algorithms. An interesting fact is that for almost all datasets KNN-based conformal predictor works best with *margin* in terms of both *oneC* and *avgC*. This pattern was not observed only for `iris` and `wave` datasets as all nonconfomity functions result in the same values of *oneC* and *avgC* in case KNN is used. This observation suggests that some classification algorithms and datasets might *prefer* particular nonconformity functions.

***Inverse probability* is almost never the best nonconformity function**. We observed that *margin* can result in the best conformal classifiers in terms of both efficiency metrics. However, it almost never happens with *inverse probability* function. In our experiments, this was observed only for Ada classifier for `user` dataset with $\epsilon = 0.15$ and `wave` dataset with $\epsilon = 0.01$.

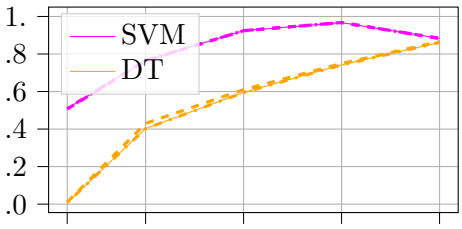
IP_M* improves *IP. In none of our experiments, we observed *IP_M* being outperformed by *IP*. *IP_M* improves *oneC* and *avgC* as compared to *IP* or produces the same values of these metrics. This is expected, as *IP_M* is basically an *IP* measure with some non-singleton predictions replaced with singletons. This replacement naturally increases *oneC* and decreases *avgC*. The fact that *IP_M* also results in valid predictions respecting the imposed value of maximum error rate ϵ, as was demonstrated in Table 3, proves the utility of this approach. Additionally, in some cases *IP_M* produces better results than both *margin* and *inverse probability* in terms of both efficiency metrics. This was observed for `glass` dataset with MPR, and for `cars` and `wave` datasets with RF.



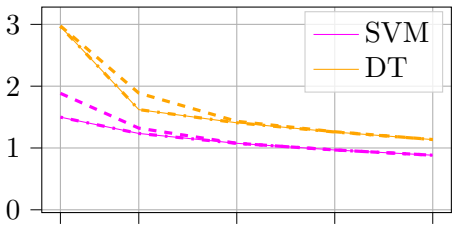
(*a*) Distribution between classes



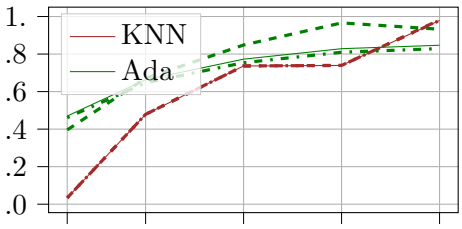
(*b*) Baseline error, *b_err*



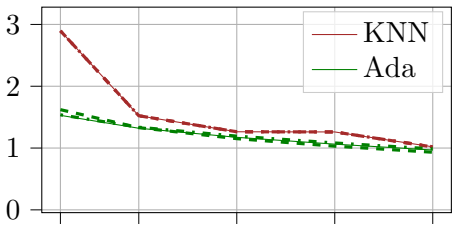
(*c*) *oneC*: SVM and DT



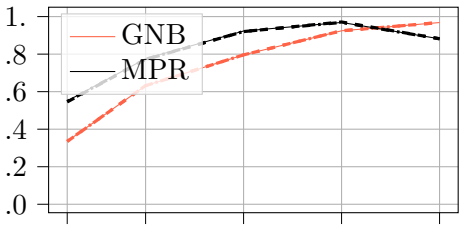
(*d*) *avgC*: SVM and DT



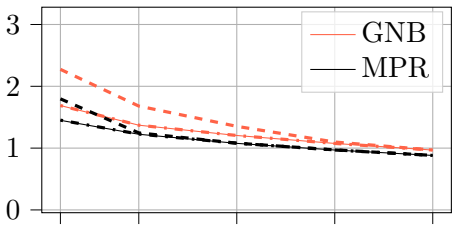
(*e*) *oneC*: KNN and Ada



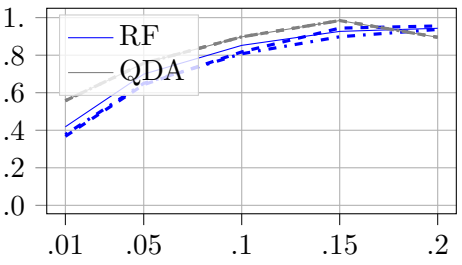
(*f*) *avgC*: KNN and Ada



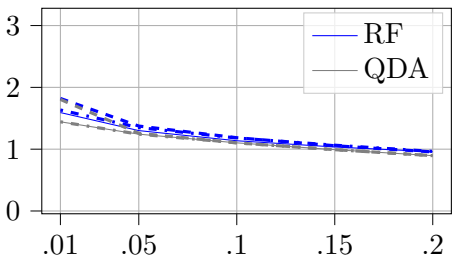
(*g*) *oneC*: GNB and MPR



(*h*) *avgC*: GNB and MPR



(*i*) *oneC*: RF and QDA

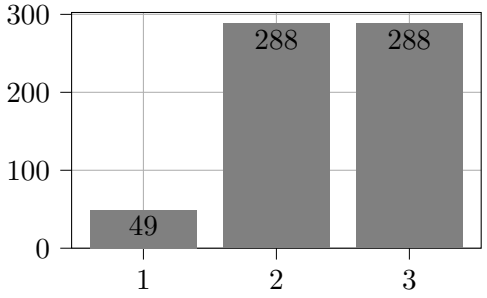


(*j*) *avgC*: RF and QDA

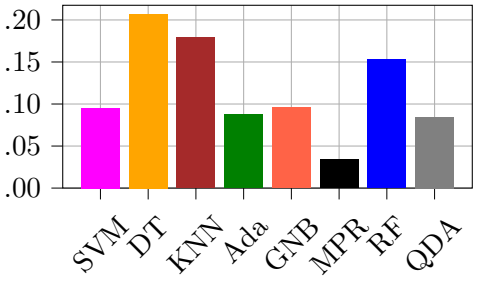
Figure 3: Results for *wave*: *M* - dashed line, *IP* - dash and dot line, *IP_M* - thin solid line

Table 5: Significance of results for the *wave* dataset

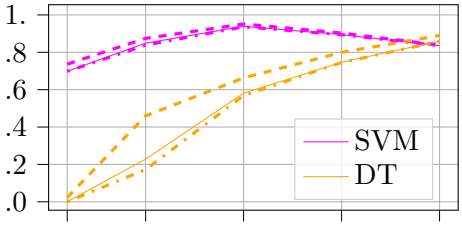
		$\epsilon = 0.01$			$\epsilon = 0.05$			$\epsilon = 0.1$			$\epsilon = 0.15$			$\epsilon = 0.2$		
oneC	ip															
	ip_m															
	m															
SVM		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip			$+^*$			$+^*$									
	ip_m			$+^*$			$+^*$									
	m	$-^*$	$-^*$		$-^*$	$-^*$										
oneC	ip						$-^*$			$-^*$						
	ip_m						$-^*$			$-^*$						
	m				$+^*$	$+^*$		$+^*$	$+^*$							
DT		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip						$+^*$			$+^*$						
	ip_m						$+^*$			$+^*$						
	m				$-^*$	$-^*$		$-^*$	$-^*$							
oneC	ip															
	ip_m															
	m															
KNN		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip															
	ip_m															
	m															
oneC	ip			$+^*$		$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$
	ip_m			$+^*$	$+^*$			$+^*$		$-^*$	$+^*$		$-^*$	$+^*$		$-^*$
	m	$-^*$	$-^*$		$+^*$			$+^*$	$+^*$		$+^*$	$+^*$		$+^*$	$+^*$	
Ada		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip			$+^*$		$-^*$			$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$
	ip_m			$+^*$	$+^*$			$+^*$		$-^*$	$+^*$		$-^*$	$+^*$		$-^*$
	m	$-^*$	$-^*$					$+^*$	$+^*$		$+^*$	$+^*$		$+^*$	$+^*$	
oneC	ip															
	ip_m															
	m															
GNB		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip			$+^*$			$+^*$			$+^*$						
	ip_m			$+^*$			$+^*$			$+^*$						
	m	$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$							
oneC	ip															
	ip_m															
	m															
MPR		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip			$+^*$												
	ip_m			$+^*$			$+^*$									
	m	$-^*$	$-^*$			$-^*$										
oneC	ip		$-^*$			$-^*$			$-^*$			$-^*$	$-^*$		$-^*$	$-^*$
	ip_m	$+^*$		$+^*$	$+^*$		$+^*$	$+^*$		$+^*$	$+^*$		$-^*$	$+^*$		$-^*$
	m		$-^*$			$-^*$			$-^*$		$+^*$	$+^*$		$+^*$	$+^*$	
RF		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip		$-^*$	$+^*$		$-^*$	$+^*$		$-^*$			$-^*$				
	ip_m	$+^*$		$+^*$	$+^*$		$+^*$	$+^*$		$+^*$	$+^*$		$+^*$			
	m	$-^*$	$-^*$		$-^*$	$-^*$			$-^*$			$-^*$				
oneC	ip															
	ip_m															
	m															
QDA		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip			$+^*$												
	ip_m			$+^*$												
	m	$-^*$	$-^*$													



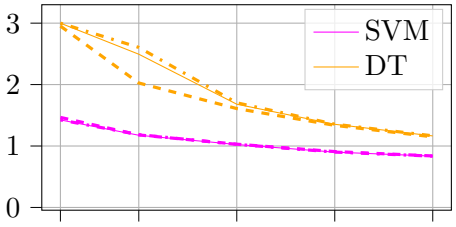
(*a*) Distribution between classes



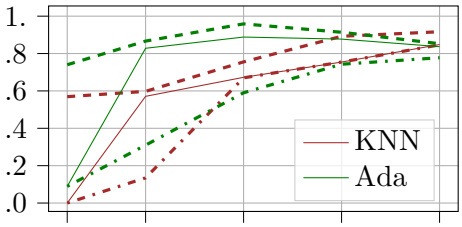
(*b*) Baseline error



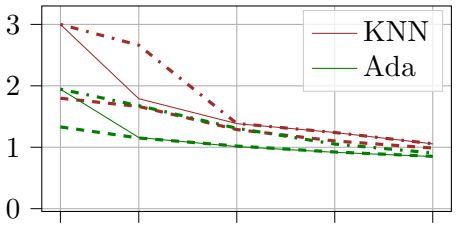
(*c*) *oneC*: SVM and DT



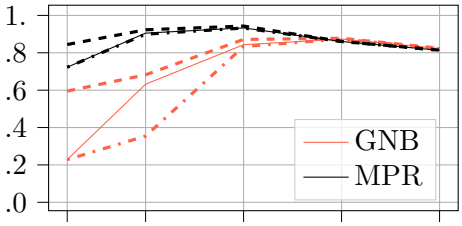
(*d*) *avgC*: SVM and DT



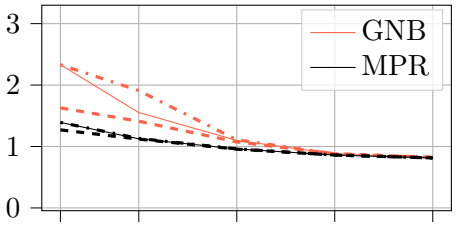
(*e*) *oneC*: KNN and Ada



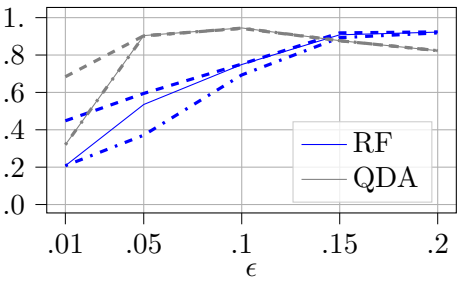
(*f*) *avgC*: KNN and Ada



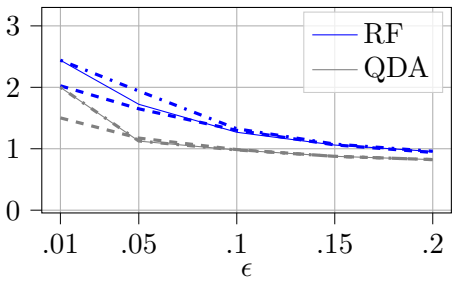
(*g*) *oneC*: GNB and MPR



(*h*) *avgC*: GNB and MPR



(*i*) *oneC*: RF and QDA



(*j*) *avgC*: RF and QDA

Figure 4: Results for *balance*: *M* - dashed line, *IP* - dash and dot line, *IP_M* - thin line

Table 6: Significance of results for the *balance* dataset

		$\epsilon = 0.01$			$\epsilon = 0.05$			$\epsilon = 0.1$			$\epsilon = 0.15$			$\epsilon = 0.2$		
oneC	ip			$-^*$			$-^*$									
	ip_m			$-^*$			$-^*$									
	m	$+^*$	$+^*$		$+^*$	$+^*$										
SVM		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip			$+^*$												
	ip_m			$+^*$												
	m	$-^*$	$-^*$													
oneC	ip			$-$		$-$	$-^*$			$-^*$			$-^*$			$-^*$
	ip_m			$-$	$+$		$-^*$			$-^*$			$-^*$			$-$
	m	$+$	$+$		$+^*$	$+^*$		$+^*$	$+^*$		$+^*$	$+^*$		$+^*$	$+$	
DT		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip					$-$	$-^*$			$-^*$						
	ip_m				$+$		$-^*$			$-^*$						
	m				$+^*$	$+^*$		$+^*$	$+^*$							
oneC	ip			$-^*$		$-^*$	$-^*$			$-^*$			$-^*$			$-^*$
	ip_m			$-^*$	$+^*$		$-^*$			$-^*$			$-^*$			$-^*$
	m	$+^*$	$+^*$		$+^*$	$+^*$		$+^*$	$+^*$		$+^*$	$+^*$		$+^*$	$+^*$	
KNN		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip			$-^*$		$-^*$	$-^*$			$-^*$			$-^*$			$-^*$
	ip_m			$-^*$	$+^*$		$-^*$			$-^*$			$-^*$			$-^*$
	m	$+^*$	$+^*$		$+^*$	$+^*$		$+^*$	$+^*$		$+^*$	$+^*$		$+^*$	$+^*$	
oneC	ip			$-^*$		$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$
	ip_m			$-^*$	$+^*$		$-^*$	$+^*$		$-^*$	$+^*$		$-^*$	$+^*$		
	m	$+^*$	$+^*$		$+^*$	$+^*$		$+^*$	$+^*$		$+^*$	$+^*$		$+^*$		
Ada		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip			$-^*$		$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$
	ip_m			$-^*$	$+^*$			$+^*$			$+^*$			$+^*$		
	m	$+^*$	$+^*$		$+^*$			$+^*$			$+^*$			$+^*$		
oneC	ip			$-^*$		$-^*$	$-^*$			$-$						
	ip_m			$-^*$	$+^*$		$-^*$			$-$						
	m	$+^*$	$+^*$		$+^*$	$+^*$		$+$	$+$							
GNB		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip			$-^*$		$-^*$	$-^*$									
	ip_m			$-^*$	$+^*$		$-^*$									
	m	$+^*$	$+^*$		$+^*$	$+^*$										
oneC	ip			$-^*$			$-$									
	ip_m			$-^*$												
	m	$+^*$	$+^*$		$+$											
MPR		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip			$-^*$												
	ip_m			$-^*$												
	m	$+^*$	$+^*$													
oneC	ip			$-^*$		$-^*$	$-^*$		$-$	$-$			$-$			
	ip_m			$-^*$	$+^*$		$-^*$	$+$								
	m	$+^*$	$+^*$		$+^*$	$+^*$		$+$			$+$					
RF		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip			$-^*$		$-^*$	$-^*$									
	ip_m			$-^*$	$+^*$		$-$									
	m	$+^*$	$+^*$		$+^*$	$+$										
oneC	ip			$-^*$												
	ip_m			$-^*$												
	m	$+^*$	$+^*$													
QDA		ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m	ip	ip_m	m
avgC	ip			$-^*$			$+^*$									
	ip_m			$-^*$			$+^*$									
	m	$+^*$	$+^*$		$-^*$	$-^*$										

The baseline pattern holds for the majority of the cases. In our experimental results, we discussed only the cases which deviate from the baseline pattern. As we saw, such cases do exist, and a particular nonconformity function produces the best values of both metrics for some baseline classifiers or some datasets. However, in most of the cases when the difference between nonconformity functions is observed, *margin* results in better values of *oneC* and *inverse probability* results in better values of *avgC*. Also, we never observed an inverse pattern, that is *inverse probability* resulting in higher values of *oneC* and *margin* resulting in lower values of *avgC* at the same time. This supports the main finding of the original paper by Johansson et al. (2017).

oneC **is not always useful.** As was discussed in Section 5.2, metric *oneC* can be misleading. For some of the datasets, only half of the singleton predictions contain the true label. In such cases, the minimization of *avgC* is preferred over the maximization of *oneC*. We also showed that the fraction of correct singleton predictions strongly correlates with the performance of the chosen classifier in the baseline scenario. It means that by analyzing this performance, we can estimate how accurate the singleton predictors will be and we can decide which efficiency metric should be considered more important. Also, it was shown that the choice of nonconformity function has little impact on the fraction of correct singleton predictions *E_oneC*.

The baseline performance of the chosen classifier impacts the efficiency of the conformal predictor. In our experiments, we observed that if the performance of the baseline classifier is good, then the choice of nonconformity function tends to have no impact on the efficiency of the resulting conformal classifier. This is the case for `iris` and `wave` datasets. In the case of `balance` dataset, this relationship is less prominent and *margin* always outperforms *IP* and *IP_M*. This can be related to the fact that this dataset is unbalanced as opposed to the two other datasets. In this case, the classifier can have problems generating predictions for the minority classes. This will not be reflected in the baseline error due to the little number of instances in these classes. The baseline performance of the underlying classification model also has a direct impact on the efficiency of the resulting conformal classifier. Except for some cases, soon after the value of ϵ reaches the value of *b_err*, metric *oneC* reaches its maximum and starts decreasing. At the same time, the value of *avgC* reaches 1 and further decreases, see, for example, results for `iris` dataset presented in Fig. 1. This observation makes sense. When $\epsilon > b\_err$, the conformal classifier is allowed to make more mistakes than it does in the baseline scenario. This can be only achieved by generating empty predictions. For such values of ϵ, more and more predictions will be singletons or empty what results in the decrease of *oneC* and *avgC* below 1.

6. Conclusions and Future Work

The objective of this paper is to further extend the recent results presented by Johansson et al. (2017) stating that there is a relationship between different model-agnostic nonconformity functions and the values of *oneC* and *avgC*. Through an empirical evaluation with ANN-based conformal predictors, the authors showed that the usage of *margin* nonconformity function results in higher values of *oneC* and *inverse probability* nonconformity function allows to achieve lower values of *avgC*. Next, it is up to the user to decide which

metric should be preferred and to choose an appropriate nonconformity function. We aim to check if the same pattern would be observed for other classification algorithms. Through experimental evaluation we showed that the previously observed pattern is supported in most of the cases, however, some classifiers and/or datasets clearly 'prefer' *margin*. This was observed for KNN classifier and `balance` dataset. At the same time, *inverse probability* is the best choice of nonconformity function only in a very small number of cases. This can be considered an exception rather than a rule.

We also proposed a method to combine *margin* and *inverse probability* into a model that we denote by *IP_M*. We showed that *IP_M* can be considered as an improved version of conformal predictor based on *IP* making this approach preferable for minimization of *avgC* in most of the cases. Additionally, it was shown that *IP_M* can be the best model in terms of both *oneC* and *avgC* in some cases: MPR-based conformal predictors for `glass` dataset, and RF-based conformal predictors for `cars` and `wave`. The validity of this approach was confirmed experimentally.

Finally, we studied how the effectiveness of the baseline classification algorithm on the given dataset can impact the efficiency of the related conformal predictor. In particular, we showed that a fraction of singleton predictions that contain the true label correlates strongly with the baseline accuracy. This observation suggests that *oneC* metric can be misleading in the case of a poorly performing baseline classifier. Our experiments also demonstrate that usually classification algorithms with higher values of baseline accuracy result in more efficient conformal predictors.

Some directions for future work are the following. It can be interesting to confirm that KNN-based conformal predictors work better with *margin* nonconformity function in the case of other datasets. Further, we would like to study which characteristics of baseline classifiers and datasets make them work better with a particular nonconformity function. For example, for `balance` dataset SVM-based conformal predictor is the only one that does not 'prefer' *margin* and follows the originally observed pattern. Next, we would like to investigate if the proposed method for combining *margin* and *inverse probability* can be improved using the latest results in assembling conformal predictors such as in Toccaceli (2019).

Acknowledgments

This work was partially supported by the European Union Horizon 2020 research programme within the project CITIES2030 "Co-creating resilient and sustainable food towards FOOD2030", grant 101000640.

References

Vineeth Nallure Balasubramanian, R Gouripeddi, Sethuraman Panchanathan, J Vermillion, A Bhaskaran, and RM Siegel. Support vector machine based conformal predictors for risk of complications following a coronary drug eluting stent procedure. In *2009 36th Annual Computers in Cardiology Conference (CinC)*, pages 5–8. IEEE, 2009.

Giovanni Cherubin. Majority vote ensembles of conformal predictors. *Machine Learning*, 108(3):475–488, 2019.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Niharika Gauraha and Ola Spjuth. Synergy conformal prediction. 2018.

Ulf Johansson, Henrik Linusson, Tuve Löfström, and Henrik Boström. Model-agnostic nonconformity functions for conformal classification. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2072–2079. IEEE, 2017.

Henrik Linusson. *Nonconformity Measures and Ensemble Strategies: An Analysis of Conformal Predictor Efficiency and Validity*. PhD thesis, Department of Computer and Systems Sciences, Stockholm University, 2021.

Henrik Linusson, Ulf Johansson, and Henrik Boström. Efficient conformal predictor ensembles. *Neurocomputing*, 397:266–278, 2020.

Harris Papadopoulos. Inductive conformal prediction: Theory and application to neural networks. In *Tools in artificial intelligence*. Citeseer, 2008.

Kostas Proedrou, Ilia Nouretdinov, Volodya Vovk, and Alex Gammerman. Transductive confidence machines for pattern recognition. In *European Conference on Machine Learning*, pages 381–390. Springer, 2002.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

Paolo Toccaceli. Conformal predictor combination using neyman–pearson lemma. In *Conformal and Probabilistic Prediction and Applications*, pages 66–88. PMLR, 2019.

Paolo Toccaceli and Alexander Gammerman. Combination of conformal predictors for classification. In *Conformal and Probabilistic Prediction and Applications*, pages 39–61. PMLR, 2017.

Paolo Toccaceli and Alexander Gammerman. Combination of inductive mondrian conformal predictors. *Machine Learning*, 108(3):489–510, 2019.

Vladimir Vovk. Transductive conformal predictors. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 348–360. Springer, 2013.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

Yachong Yang and Arun Kumar Kuchibhotla. Finite-sample efficient conformal prediction. *arXiv preprint arXiv:2104.13871*, 2021.