Journal of Machine Learning Research 1–28 Workshop on Distribution-Free Uncertainty Quantification at ICML 2021

How Nonconformity Functions and Difficulty of Datasets Impact the Efficiency of Conformal Classifiers

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

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 systems can output multiple class labels instead of one. It is also known from the literature, 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. The experimental evaluation is done using 11 real and 5 synthetic datasets.

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 the nonconformity function known as *margin* results in higher *oneC* and the usage of *inverse probability* (also known as *hinge*) as a nonconformity function results in lower values of *avgC*. In the rest of the text, we will refer to this relationship as a baseline or original pattern (relationship). The authors use 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 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, 5 synthetic datasets and 11 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 the 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 characteristics.

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 functions have 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 works. 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 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 knows 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, the *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 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. To our opinion, 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 result 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*. 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 Sections 5.1.2 and 5.2.2 and its efficiency is demonstrated in Sections 5.1.4 and 5.2.4.

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 5.2.3, 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 presented in Fig. 1. **First**, we construct conformal predictors using both nonconformity functions separately[1]. 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 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.

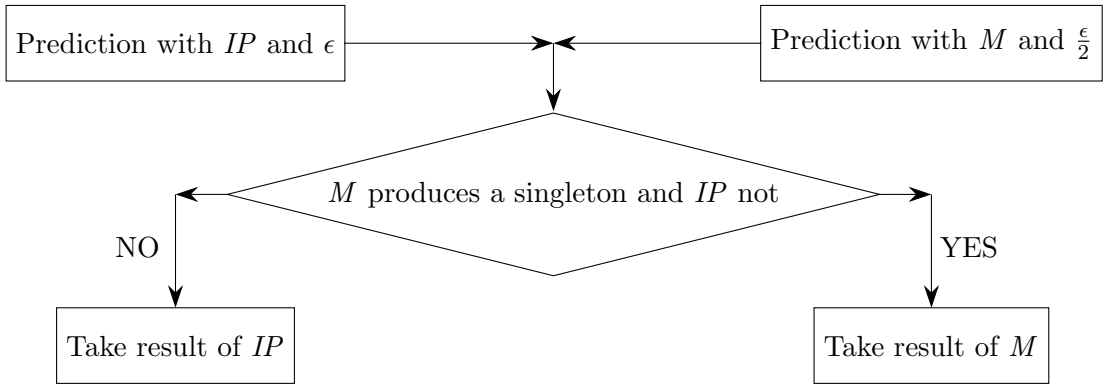


Figure 1: Algorithm for combining *margin* (M) and *inverse probability* (IP): IP_M.

In the rest of the paper, we use M and IP to refer to the conformal classifiers based on *margin* and *inverse probability* respectively. IP_M will be 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.

1. See Vovk et al. (2005); Shafer and Vovk (2008); Johansson et al. (2017) for an explanation of how conformal predictors are constructed.

4. Experimental setup

To perform experimental analysis, we used the implementation of conformal predictors available from *nonconformist*[2] 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, i.e., 80% of the training set was used for actual training of the classification model, and the rest 20% were used for calibration. 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 the UCI repository Dua and Graff (2017). In this paper, we present not aggregated, but detailed results for every analyzed dataset. That is why we chose 11 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 9. Additionally, we aim to analyze the impact of the 'easiness' of a dataset on the performance of conformal predictors. For this, we generate synthetic datasets of different difficulties. The characteristics of these datasets are presented in the first section of Table 2 and are discussed in Section 5.1. We start with the analysis of the results for synthetic datasets in Section 5.1. After that, we proceed to the analysis of results obtained for real-world datasets in Section 5.2.

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 Vector 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 the *scikit-learn* Python library. In Table 1, we summarize the input parameters of these algorithms unless the default values are used.

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

Table 1: Input parameters of classification algorithms

Different classifiers perform differently on different datasets. We demonstrate this with the error in the baseline mode[3] *b_err* in the first section of Tables 2 and 9. As will be discussed later, 8 classifiers perform similarly in terms of baseline error on synthetic datasets and produce different results on real-world datasets. That is why in Table 9 we report both

2. https://github.com/donlnz/nonconformist
3. In this text, we use the *baseline mode* to refer to the standard (non-conformal) prediction.

the range of *b_err* and the median values for real datasets. However, for the generated datasets, we report only the median of *b_err* in Table 2. 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.

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 a 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.

5. Experimental results

In this section, we present experimental results for synthetic (see Section 5.1) and real-world (see Section 5.2) datasets. The demonstrated results can be reproduced using Python code from the relevant repository[4].

5.1. Synthetic datasets

5.1.1. Description

To study the impact of the 'easiness' of the dataset on the performance of conformal classifiers with different nonconformity functions, we generated 5 artificial datasets. All 5 datasets have 2 attributes and 4 classes with 2000 instances per class, see section 1 of Table 2. The 4 classes of every dataset are defined as a set of normally distributed points around 4 centers on a 2D plane: $(1, 0)$, $(0, 1)$, $(-1, 0)$ and $(0, -1)$. The difference between the 5 datasets is in the value of standard deviation σ used to generated the normally distributed points. The latter has 5 possible values: $\sigma \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. The impact of the value of σ on the distribution of points is visually demonstrated in Figs. 2(a), 3(a), 4(a), 5(a) and 6(a). As expected, the datasets with a larger value of σ are also more difficult to classify, see median value *b_err* in section 1 of Table 2 and the values of *b_err* per classifier in Figs. 2(b), 3(b), 4(b), 5(b) and 6(b). In the rest of the text, we use the corresponding value of σ to refer to different synthetic datasets.

5.1.2. 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 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 for every synthetic dataset and an average over them. As we can see, all conformal predictors are well-calibrated. The only exception is the $\sigma = 0.2$ dataset for which the empirical error rates are usually lower than the value of ϵ. This difference is the most prominent in the case of large values of significance (for $\epsilon = 0.15$ and $\epsilon = 0.2$). This can be explained by the fact that the 4 classes of the synthetic dataset

4. https://github.com/marharyta-aleksandrova/copa-2021-conformal-learning

Section		**Datasets**	$\sigma = 0.2$	$\sigma = 0.4$	$\sigma = 0.6$	$\sigma = 0.8$	$\sigma = 1.0$
1	info	# instances	8000				
		# attributes	2				
		# classes	4				
		balanced	Y				
		b_err median, %	0.001	0.081	0.227	0.338	0.429
2	*E_oneC*	mean	1.0	0.949	0.862	0.828	0.775
		mean-std	0	0.002	0.018	0.018	0.025
		corr. *b_acc*	0.936	0.974	0.936	0.532	0.734
3	stat. thres.	*oneC*,%	0	37.5	80	92.5	92.5
		avgC,%	0	27.5	60	77.5	97.5
		oneC,%	7.5	37.5	90	95	97.5
		avgC,%	7.5	35	82.5	95	97.5
4	pattern	inverse pattern	GNB [0.2], RF [0.2], QDA [0.2]				
		M is the best		**KNN** [0.05], Ada, RF [0.1]	**KNN** [0.05, 0.15], Ada, GNB [0.2], MPR [0.2], QDA [0.2]	SVM [0.01], Ada [0.05, 0.1], **KNN** ~~(0.01, 0.15)~~	SVM [0.01], **KNN** ~~(0.01)~~
		IP is the best			KNN [0.1]		
		$IP\_M \geq IP$	Y	Y	Y	Y	Y

Table 2: Synthetic datasets: Characteristics of the datasets and summarization of results. ~~Striketrough~~ text indicates those values of ϵ, for which we do not observe a particular pattern.

with $\sigma = 0.2$ are well-separated, see Fig. 2(a), and allow almost perfect classification, see the values of *b_err* in Table 2 and in Fig. 2(b). This also affects the average values presented in the section **MEAN** of Table 3.

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.

$\sigma = 0.2$	ϵ	*IP*	*IP_M*	*M*	$\sigma = 0.8$	ϵ	*IP*	*IP_M*	*M*
	0.01	0.01	0.01	0.01		0.01	0.01	0.01	0.01
	0.05	0.04	0.04	0.04		0.05	0.04	0.05	0.05
	0.10	0.08	0.08	0.08		0.10	0.09	0.10	0.10
	0.15	0.11	0.11	0.11		0.15	0.15	0.15	0.14
	0.20	0.13	0.13	0.14		0.20	0.19	0.20	0.20
$\sigma = 0.4$	ϵ	*IP*	*IP_M*	*M*	$\sigma = 1.0$	ϵ	*IP*	*IP_M*	*M*
	0.01	0.01	0.01	0.01		0.01	0.01	0.01	0.01
	0.05	0.05	0.05	0.05		0.05	0.04	0.05	0.05
	0.10	0.10	0.10	0.10		0.10	0.09	0.10	0.10
	0.15	0.14	0.15	0.15		0.15	0.14	0.16	0.15
	0.20	0.19	0.20	0.19		0.20	0.19	0.20	0.20
$\sigma = 0.6$	ϵ	*IP*	*IP_M*	*M*	**MEAN**	ϵ	*IP*	*IP_M*	*M*
	0.01	0.01	0.01	0.01		0.01	0.01	0.01	0.01
	0.05	0.04	0.05	0.05		0.05	0.04	0.05	0.05
	0.10	0.10	0.10	0.09		0.10	0.09	0.10	0.09
	0.15	0.14	0.15	0.14		0.15	0.14	0.14	0.14
	0.20	0.19	0.20	0.19		0.20	0.18	0.19	0.18

Table 3: Synthetic datasets: Empirical error rates

5.1.3. 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 2 of Table 2.

The first row of this section shows the averaged value of *E_oneC* overall 5 values of ϵ and 3 nonconformity functions. We can notice that this value decreases when the difficulty of a dataset increases. It drops from 1.0 for $\sigma = 0.2$ to 0.775 for $\sigma = 1.0$. This means that on average more than 20% of the produced singleton predictions for the latter dataset do not contain the true label.

To further analyze the relationship between *E_oneC* and *b_err*, we calculated the value of correlation between the corresponding characteristics through 8 baseline classifiers within the results for a particular dataset. The results are presented in the third row *corr. b_acc*. We can see that the correlation is always high and exceeds 0.9 for 3 of 5 datasets. The lowest value of 0.532 is observed for $\sigma = 0.8$. 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.

5.1.4. 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 in figures like Figs. 2 and 3. Such figures contain visualization of 4 defined 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 presented in tables like Tables 4 and 5. 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[5]. To avoid too small differences, we put a sign into the matrix only if the corresponding difference is above the threshold of 2%[6] or it is statistically significant. Therefore, an empty cell indicates that neither threshold differences of at least 2%, nor statistically significant differences were observed for the relative pair of nonconformity functions. For example, from Table 5 we can see that *margin* results in better values of *oneC* than *IP* and *IP_M* for SVM with $\epsilon = 0.01$. These results are also statistically significant, as indicated by a *. Section 3 of Table 2 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 tables like Table 5. 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. By analyzing the corresponding values from section

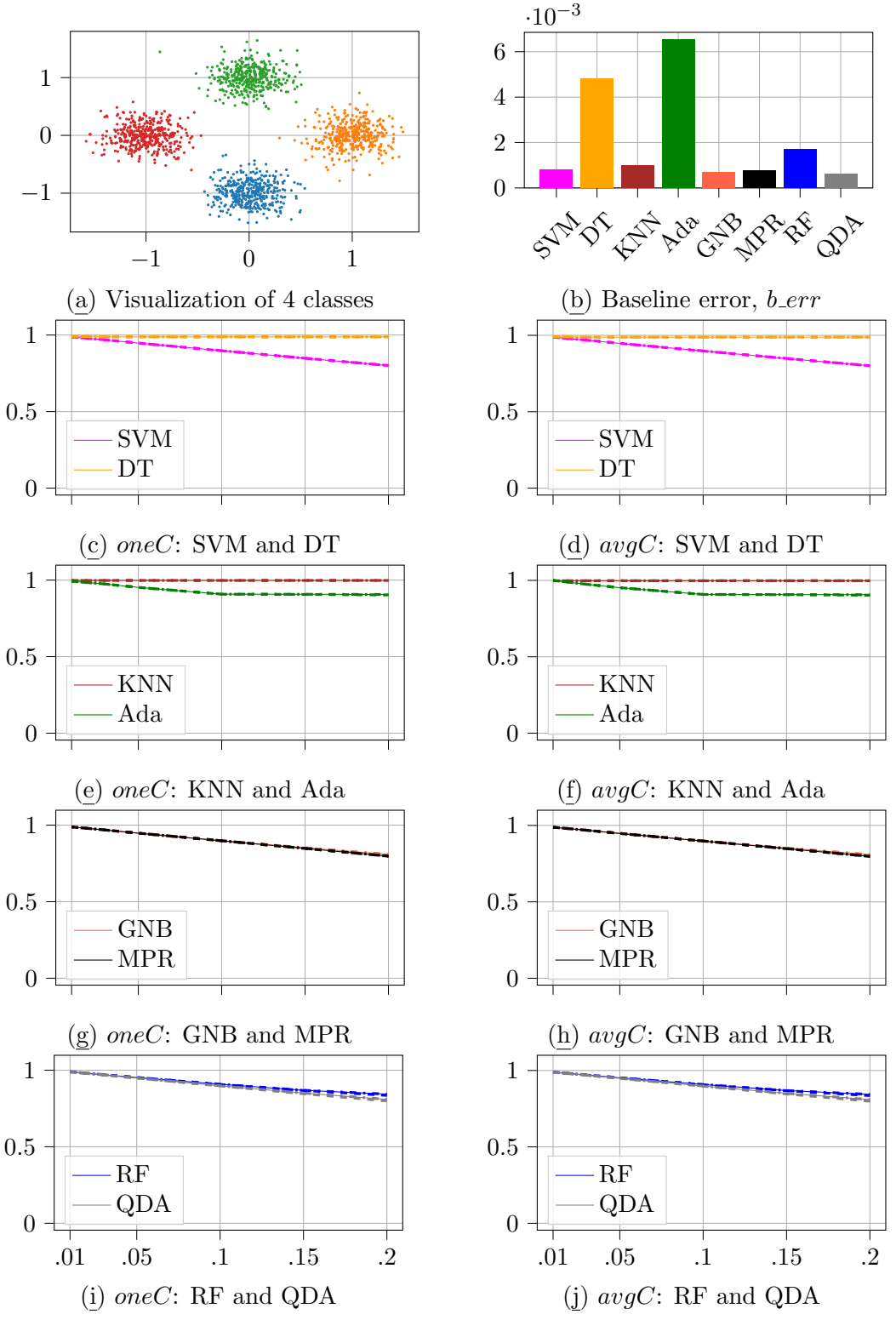
5. Statistical significance was estimated using Student's t-test with $\alpha = 0.05$.
6. For 100% we take the value of 1 for *oneC* and the total number of classes for *avgC*. These are the maximum values of these two metrics.

3 of Table 2, we can notice that for all synthetic datasets we observe more statistically significant differences than differences exceeding the threshold of 2%. Further analysis of Tables 4 to 8 shows that all observed differences for synthetic datasets are statistically significant.

Using the information provided in figures like Figs. 2 and 3 and tables like Tables 4 and 5, we can analyze the efficiency of conformal classifiers for different nonconformity functions and identify which nonconformity functions perform better. The corresponding findings are summarized in section 4 of Table 2. This section shows the deviations from the pattern originally observed by Johansson et al. (2017). In our experiments with synthetic datasets, we observed the following 3 deviations: 1) **inverse pattern**: opposite to the pattern observed in the original study, *inverse probability* results in higher values of *oneC* and *margin* results in lower values of *avgC*; 2) ***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; 3) ***IP* is the best**: *inverse probability* is the best choice of nonconformity function. Additionally, our experiments show that *IP_M* never performs worse than *IP* (*IP_M* $\geq$ *IP*). In the rest of this section, we analyze in detail the results for 5 synthetic datasets. The general conclusions are discussed in Section 5.3.

The detailed results for the synthetic dataset with $\sigma = 0.2$ are presented in Fig. 2 and Table 4. As shown in section 3 of Table 2, we observe no threshold differences between nonconformity functions, and the statistically significant difference is observed only in 7.5% of setups. This is also reflected in the corresponding plots for *oneC* and *avgC* in Fig. 2. That is why we demonstrate only a part of Table 4 that corresponds to those baseline classifiers, for which statistically significant differences were observed. As it was mentioned above, this dataset is very easy to classify, all baseline classifiers result in *b_err* less than 1%, see Fig. 2(b). Analyzing the results for the effectiveness metrics, we can see that the plots for *oneC* and *avgC* look identical. This can be explained by the fact that the very low values of *b_err* allow achieving perfect values of *oneC* and *avgC* being equal to 1 for $\epsilon = 0.01$. Increasing the significance level further only results in decreasing of *oneC* and *avgC* below 1. This dataset is also the only one for which we observe the **inverse pattern**. This is the case for GNB, RF, and QDA with $\epsilon = 0.2$. As indicated in Table 4, these differences are also statistically significant. Given that this dataset is unrealistically easy and the relative conformal predictors are also not well-calibrated, see Table 3, this observation can be considered as an exception rather than a rule.

Fig. 3 and Table 5 demonstrate the results obtained for the next synthetic dataset with $\sigma = 0.4$. As we can see from Fig. 3(a), the instances of different classes now overlap. This also results in higher values of *b_err*, see Fig. 3(b). Most classifiers result in *b_err* below 10% with the only exception being Ada classifier with *b_err* = 0.2. Analyzing corresponding plots for *oneC* and *avgC*, we can also notice that the baseline performance correlates with the effectiveness of conformal predictors. For example, the conformal predictor based on Ada classifier results in the lowest values of *oneC* and also tends to produce higher values of *avgC* than others. For this dataset, we can observe ***M* is the best** pattern. *Margin* nonconformity function results in the best performance for KNN with $\epsilon = 0.05$, RF with $\epsilon = 0.1$ and Ada with all values of epsilon, see section 4 of Table 2. As indicated in the corresponding matrices of Table 5, the gain in performance provided by *margin* is also statistically significant. Finally, *IP_M* either improves the effectiveness as compared with



(a) Visualization of 4 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

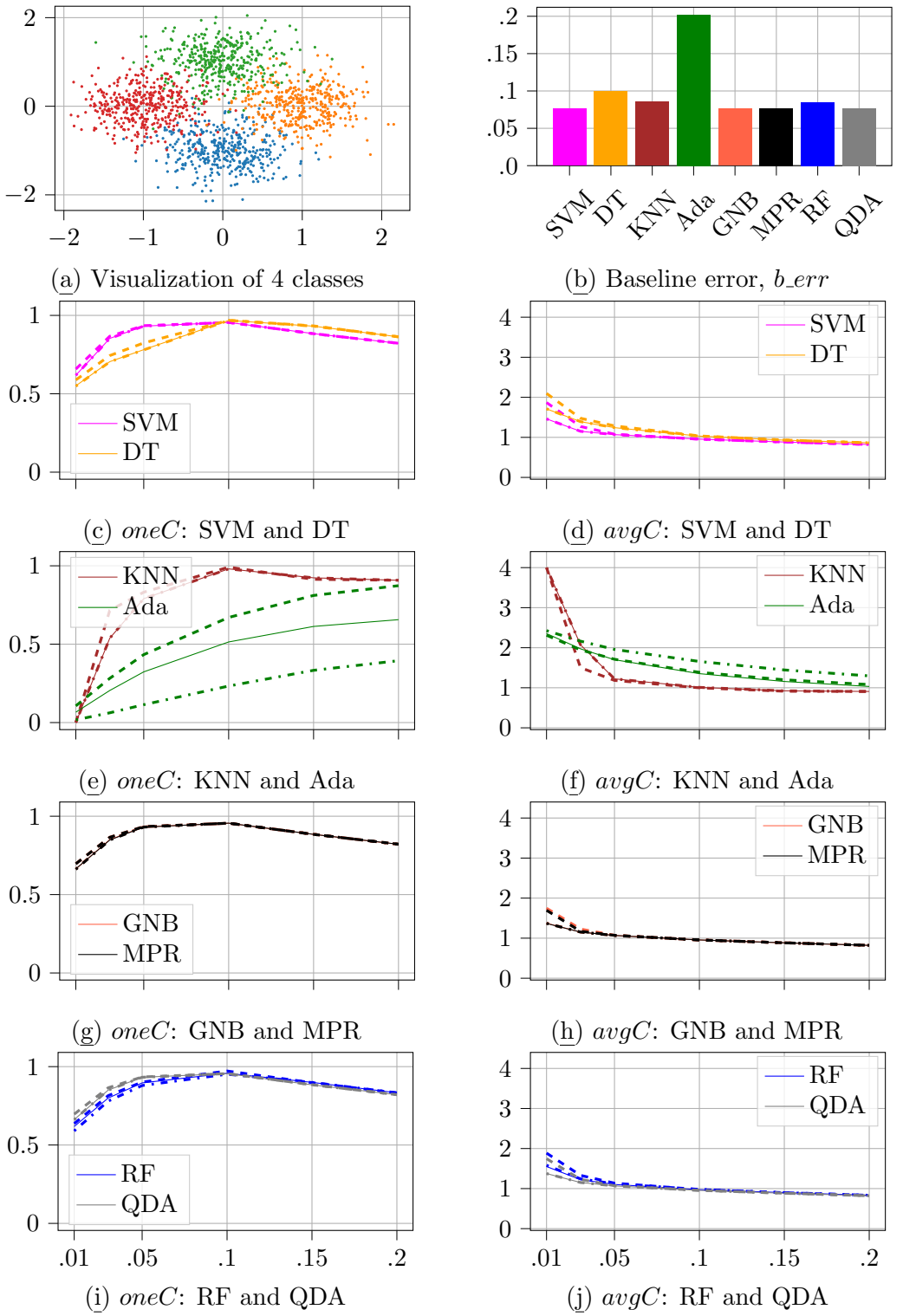
Figure 2: $\sigma = 0.2$: *M* - dashed line, *IP* - dash and dot line, *IP_M* - thin solid line.

		$\epsilon = 0.01$			$\epsilon = 0.05$			$\epsilon = 0.1$			$\epsilon = 0.15$			$\epsilon = 0.2$		
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													$-^*$	$-^*$	
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													$+^*$	$+^*$	

Table 4: Significance of results for $\sigma = 0.2$. An empty cell indicates similar performance, see Section 5.1.4.

inverse probability nonconformity function or does not change it ($IP\_M \geq IP$ pattern). In none of the cells of Table 5 *IP_M* is dominated by *IP*. This is also visually visible for the Ada classifier. As shown in Figs. 3(e) and 3(f), *IP_M* substantially increases *oneC* and decreases *avgC*. Additionally, for all values of ϵ except 0.01, *IP_M* results in the lowest *avgC* as compared to both *inverse probability* and *margin*. Finally, we can observe a decrease in the values of *oneC* after ϵ reaches the value close to the baseline error of the underlying classifier. The values of *avgC* approach 1 and then further decreases at the same time. This is observed for all classifiers except Ada in the corresponding plots of Fig. 3. It happens because we do not perform experiments for $\epsilon > 0.2$, which corresponds to *b_err* of this classifier. This observation can be explained by the fact that when $\epsilon \approx b\_err$, the conformal classifier is allowed to make as many errors as the baseline model would, thus resulting in the maximum number of singleton predictors. Further increase of the value of error rate can be achieved only at the increase of empty predictors. This results in the decrease of *oneC* and the further decrease of *avgC* below 1.

The results for the synthetic dataset with $\sigma = 0.6$ are presented in Fig. 4 and Table 6. The difficulty of the dataset increases as indicated by the visualization in Fig. 4(a) and the values of *b_err* in Fig. 4(b). Now for all conformal classifiers, we can observe a clear visual difference between nonconformity functions. As indicated in section 3 of Table 2, for more than 60% of setups the deviation is above the 2% threshold and in more than 80% the difference is statistically significant. As in the previous case, the performance in the baseline mode tends to correlate with the efficiency of the resulting conformal classifiers. This is illustrated for the 2 least accurate classifiers KNN and Ada in Figs. 4(e) and 4(f). For this



(a) Visualization of 4 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: $\sigma = 0.4$: *M* - dashed line, *IP* - dash and dot line, *IP_M* - thin solid line.

		$\epsilon = 0.01$			$\epsilon = 0.05$			$\epsilon = 0.1$			$\epsilon = 0.15$			$\epsilon = 0.2$		
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	$-^*$	$-^*$													

Table 5: Significance of results for $\sigma = 0.4$. An empty cell indicates similar performance, see Section 5.1.4.

dataset, *inverse probability* nonconformity function is the best choice for KNN with $\epsilon = 0.1$, see the corresponding matrix in Table 6. Additionally, *margin* nonconformity function is the best choice for Ada, KNN with $\epsilon \in \{0.05, 0.15\}$, GNB, MPR and QDA with $\epsilon = 0.2$.

In Fig. 5 and Table 7 we present the results for the synthetic dataset with $\sigma = 0.8$. The difficulty of the dataset increases even further with a median of *b_err* now reaching 33.8%, see Fig. 5(b). Again, we observe a correlation between the accuracy of a classifier in the baseline mode and the efficiency of the resulting conformal predictor. Less accurate classifiers produce conformal predictors with lower values of *oneC* and larger values of *avgC*, see the results for KNN and Ada in Figs. 5(e) and 5(f) for a prominent example. The difference between different nonconformity functions is visible for all algorithms. In 95% of setups the observed differences are statistically significant, see Table 2. For this dataset, we again observe ***M* is the best** pattern. This is true for SVM with $\epsilon = 0.01$, Ada with $\epsilon \in \{0.05, 0.1\}$ and KNN will all values of ϵ except 0.01 and 0.15[7]. This is visible for KNN classifier in Figs. 5(e) and 5(f) and confirmed by the corresponding matrices in Table 7.

Finally, the results for the last synthetic dataset with $\sigma = 1.0$ are presented in Fig. 6 and Table 8. From Fig. 6(a) we can see that a large portion of classes now overlap resulting in values of *b_err* surpassing 40%, see Fig. 6(b). With the increase in dataset complexity, the difference between the performance of different nonconformity functions becomes more prominent. As in previous cases, the least accurate classifiers KNN and Ada produce conformal predictors of lower efficiency. Finally, *margin* nonconformity function results in the best performance for SVM with $\epsilon = 0.01$ and KNN with all values of ϵ except 0.01, see Figs. 6(c) to 6(f) and the corresponding cells in Table 8.

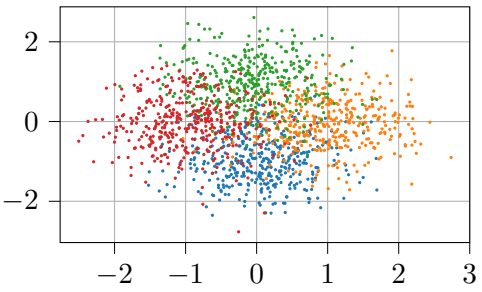
There is also an interesting trend shared by most synthetic datasets: the observed relationships stay identical for all values of ϵ for some baseline algorithms. It means that for the relative algorithms, the dominance relationship between nonconformity functions does not change with ϵ. This tendency was not observed only for the dataset with $\sigma = 0.2$, which is also an unrealistically easy dataset. This observation holds for the following cases:

- $\sigma = 0.4$: Ada classifier for both *oneC* and *avgC* - all corresponding matrices for *oneC* and *avgC* are identical in Table 5;
- $\sigma = 0.6$: Ada for *oneC* and DT for *avgC*, see Table 6;
- $\sigma = 0.8$: Ada and RF for *oneC*, see Table 7;
- $\sigma = 1.0$: DT, GNB, RF, QDA for *oneC* and MPR for both *oneC* and *avgC*, see Table 8.

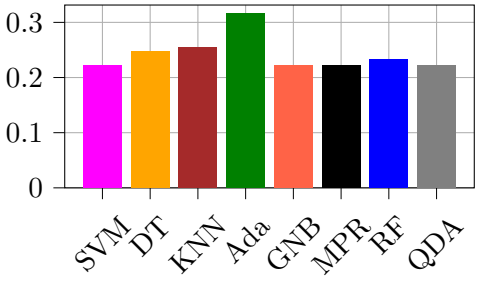
5.2. Real-world datasets

We performed the same set of experiments for the real-world datasets from Table 9 as for the synthetic datasets. To avoid redundancy and reduce the size of this paper, in this section we discuss only aggregated results presented in Table 9 and do not present plots and significance tables like Fig. 2 and Table 4 for individual datasets.

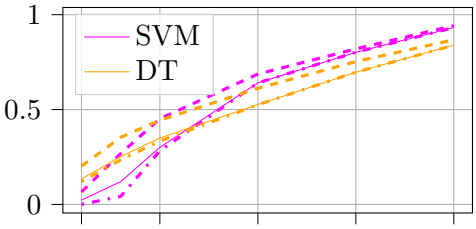
7. If a particular pattern is not observed for some small number of values of ϵ, we indicate it with strikethrough text in Tables 2 and 9.



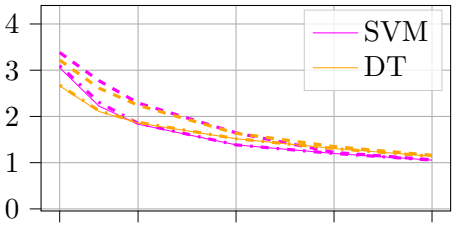
(a) Visualization of 4 classes



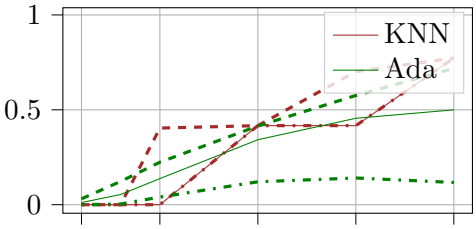
(b) Baseline error, *b_err*



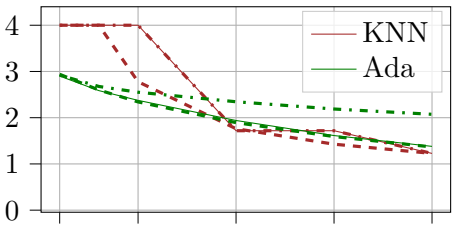
(c) *oneC*: SVM and DT



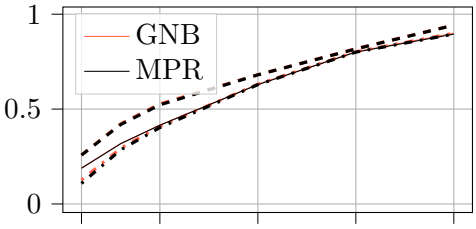
(d) *avgC*: SVM and DT



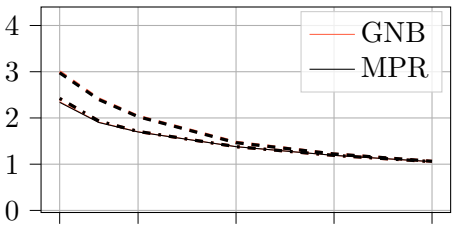
(e) *oneC*: KNN and Ada



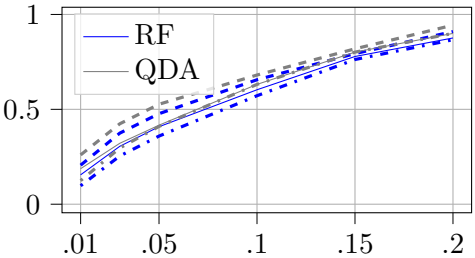
(f) *avgC*: KNN and Ada



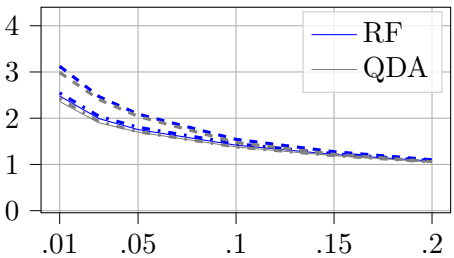
(g) *oneC*: GNB and MPR



(h) *avgC*: GNB and MPR



(i) *oneC*: RF and QDA

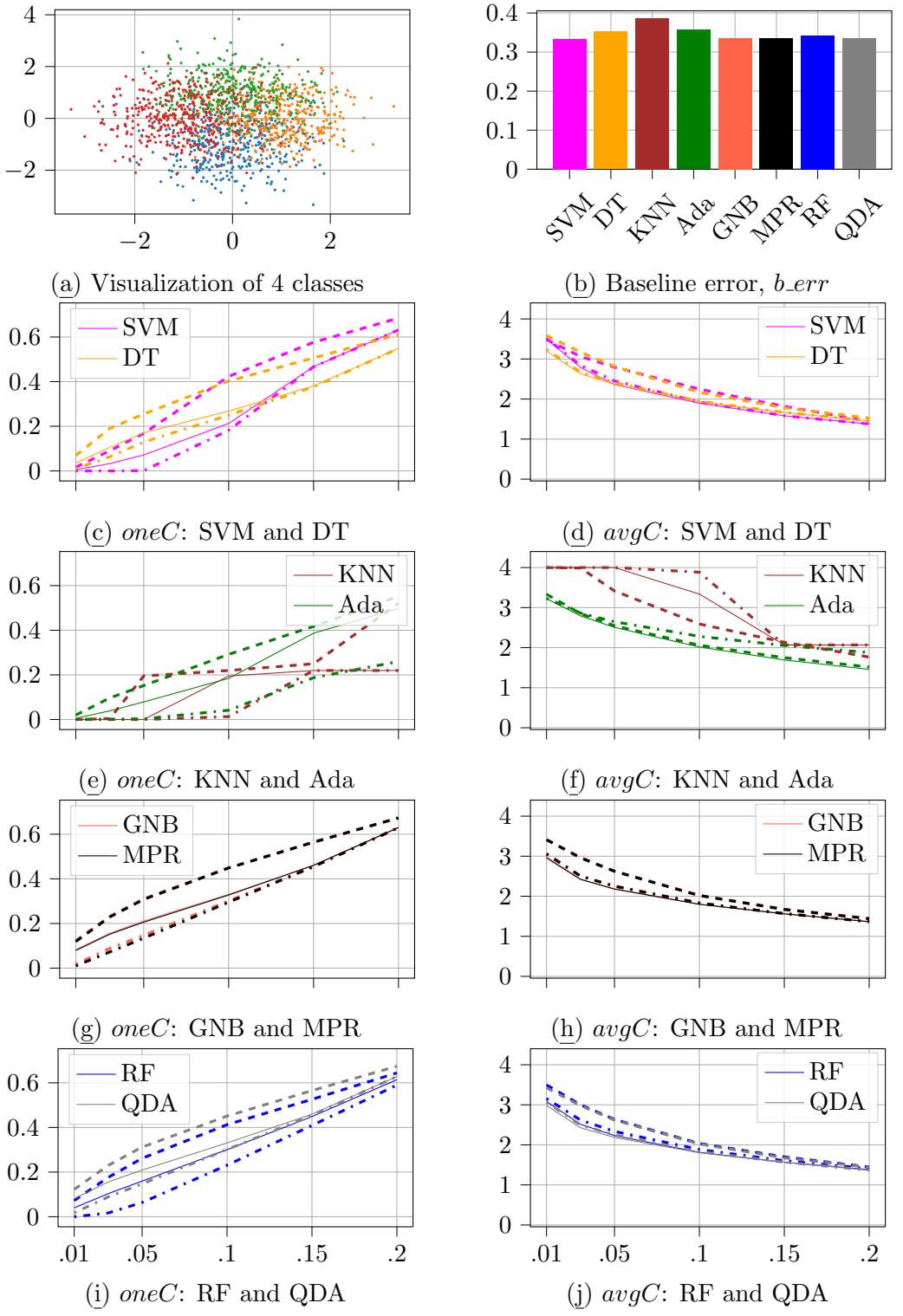


(j) *avgC*: RF and QDA

Figure 4: $\sigma = 0.6$: *M* - dashed line, *IP* - dash and dot line, *IP_M* - thin solid line.

		$\epsilon = 0.01$			$\epsilon = 0.05$			$\epsilon = 0.1$			$\epsilon = 0.15$			$\epsilon = 0.2$		
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	$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$				

Table 6: Significance of results for $\sigma = 0.6$. An empty cell indicates similar performance, see Section 5.1.4.



(a) Visualization of 4 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

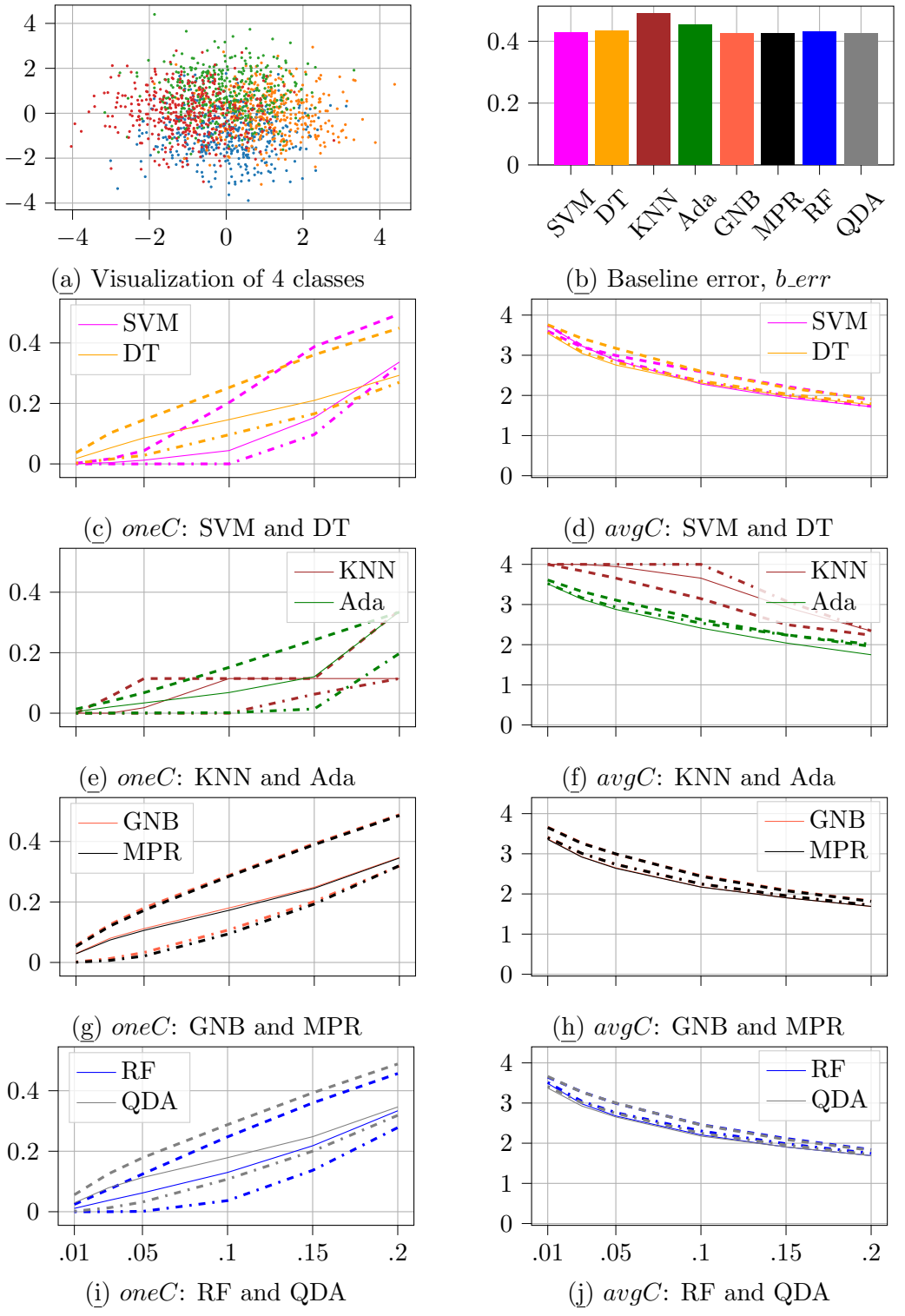
(i) *oneC*: RF and QDA

(j) *avgC*: RF and QDA

Figure 5: $\sigma = 0.8$: M - dashed line, IP - dash and dot line, IP_M - thin solid line.

		$\epsilon = 0.01$			$\epsilon = 0.05$			$\epsilon = 0.1$			$\epsilon = 0.15$			$\epsilon = 0.2$		
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	$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$	

Table 7: Significance of results for $\sigma = 0.8$. An empty cell indicates similar performance, see Section 5.1.4.



(a) Visualization of 4 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 6: $\sigma = 1.0$: M - dashed line, IP - dash and dot line, IP_M - thin solid line.

		$\epsilon = 0.01$			$\epsilon = 0.05$			$\epsilon = 0.1$			$\epsilon = 0.15$			$\epsilon = 0.2$		
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	$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$		$-^*$	$-^*$	

Table 8: Significance of results for $\sigma = 1.0$. An empty cell indicates similar performance, see Section 5.1.4.

5.2.1. Description

The general description of the 11 chosen real-world datasets is presented in section 1 of Table 9. As we can see, they vary significantly in their characteristics. The number of instances changes from 150 for iris dataset to 5000 for wave, the number of attributes changes from 4 to 21, and the number of classes from 3 to 10. Some datasets are perfectly balances, for example, iris dataset with exactly 50 instances per class, while others are highly unbalanced, for example, wineR and wineW. *b_err range* from the fifth row of section 1 in Table 9 demonstrates the range of errors produced by all 8 classification algorithms in the baseline mode. 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.

5.2.2. Validity

In Table 10 we demonstrate the empirical error rates averaged among all datasets. As we can see, all conformal predictors are well-calibrated, including the combination *IP_M*. Similar to the synthetic datasets, the relatively lower values of empirical error rates for larger values of ϵ are due to the disproportionally low values obtained for easy datasets like iris.

5.2.3. Informativeness of *oneC*

Similarly to synthetic datasets, we analyze the informativeness of *oneC* metric in section 2 of Table 9. We can notice that the average value of *E_oneC* 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, such a prediction can be more confusing than a prediction with multiple labels.

Again, there is a strong correlation between the mean value of *E_oneC* and the difficulty of the dataset for the baseline classifiers (*b_err*). The corresponding results are indicated in the third row *corr. b_acc*. We can see that for 6 of 11 datasets (55%) the correlation is around 0.9 or above. This holds for iris, user, cars, wave, yeast and cool datasets. For 2 more datasets (balance and wineW), the correlation coefficient is approximately 0.8. For glass dataset, it is equal to 0.69, for heat to 0.57 and only for wineR the correlation is as low as 0.27. These results confirm a strong relationship between the baseline error of the underlying classification model and the correctness of singleton predictions as observed in Section 5.1.3. Similarly to synthetic datasets, *mean-std* is very low, confirming that *E_oneC* does not depend on the choice of nonconformity function.

5.2.4. Efficiency of different nonconformity functions

Finally, in this section, we discuss the observed relationships between different nonconformity functions and the effectiveness of the resulting conformal predictors presented in

Section		**Datasets**	iris	user	glass	cars	wave	balance	wineR	wineW	yeast	heat	cool
1	info	# instances	150	403	214	1728	5000	625	1599	4898	1484	768	768
		# attributes	4	5	9	6	21	4	11	11	8	8	8
		# classes	3	4	6	4	3	3	6	7	10	8	8
		balanced	Y	mostly	N	N	Y	N	N	N	N	N	N
		b_err range, %	2-6	5-32	24-92	7-96	13-25	3-21	38-50	45-58	40-87	39-97	26-79
		b_err med., %	4	7	48	13	17	10	45	53	44	40	48
2	*E_oneC*	mean	0.98	0.93	0.63	0.88	0.9	0.96	0.52	0.5	0.56	0.78	0.63
		mean-std	0	0	0.02	0	0	0	0.03	0.02	0.02	0.04	0.05
		corr. *b_acc*	0.92	0.98	0.69	0.93	0.92	0.8	0.27	0.78	0.97	0.57	0.89
3	stat. thres.	*oneC*, %	5	15	75	47.5	25	67.5	72.5	65	77.5	70	60
		avgC, %	2.5	22.5	67.5	37.5	27.5	42.5	80	77.5	82.5	65	62.5
	stat.	*oneC*, %		5	50	45	30	55	82.5	85	85	62.5	62.5
		avgC, %		17.5	37.5	45	47.5	50	82.5	77.5	77.5	72.5	55
4	patterns	*M* is the best		**KNN**	**KNN**, DT	**KNN**	Ada ~~(0.01, 0.05)~~	**KNN**, RF, DT, MPR, GNP, Ada, QDA	**KNN** [0.05, 0.1], DT [0.05]	**KNN** [0.05, 0.1]	**KNN** ~~(0.01, 0.2)~~, DT [0.05], Ada [0.15, 0.2], QDA [0.15, 0.2]	**KNN** ~~(0.05)~~, GNB [0.01], RF [0.2]	**KNN** ~~(0.01)~~, GNB ~~(0.15, 0.20)~~ DT [0.01], RF [0.2], SVM [0.2],
		IP is the best		Ada [0.15]			Ada [0.01]					DT [0.01], Ada ~~(0.15, 0.2)~~	SVM [0.01, 0.05], Ada ~~(0.2)~~
		IP_M is the best			MPR	RF	RF					RF [0.05]	
		$IP\_M \geq IP$	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Table 9: Real-world datasets: Characteristics of the datasets and summarization of results. ~~Strikethrough~~ text indicates those values of ϵ, for which we do not observe a particular pattern.

ϵ	**0.01**	**0.05**	**0.10**	**0.15**	**0.20**
IP	0.01	0.04	0.09	0.14	0.18
IP_M	0.01	0.05	0.09	0.14	0.18
M	0.01	0.05	0.09	0.14	0.18

Table 10: Real-world datasets: Empirical error rates

section 4 of Table 9. First, we never observe the *inverse pattern* that was present for the synthetic dataset with $\sigma = 0.2$ for GNB, RF and QDA with $\epsilon = 0.2$. This proves that the inverse pattern indeed is rather an exception than a rule, as was discussed in Section 5.1.4. As before, for many settings *margin* is the best choice of nonconformity function. Interestingly, it is the case for almost all real-world datasets when KNN is used as a baseline classifier. Additionally, in the case of balance dataset, almost all conformal classifiers, except the one based on SVM, perform the best when *margin* is used. We observe that *inverse probability* nonconformity function can be the best choice more often than for synthetic datasets. It is observed for 4 real-world datasets: user, wave, heat, and cool. Finally, for some settings, the combination of *inverse probability* and *margin* provides better values of both *oneC* and *avgC*. This holds for user dataset with MPR and for cars, wave, and heat datasets with RF. Such a pattern was never observed for synthetic datasets.

5.3. Summary of results

In this subsection, we summarize the findings from our experimental results shown in Tables 2 and 9.

For one unrealistically easy synthetic dataset with $\sigma = 0.2$, we 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. This is the case for GNB, RF and QDA classifiers with $\epsilon = 0.2$. For this setting, none of the nonconformity functions results in empirical error rates close to the defined value of ϵ, see Table 3. This can be a possible explanation of the observed deviation.

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, wave and the synthetic dataset with $\sigma = 0.1$. In these cases, all nonconformity functions result in the same values of *oneC* and *avgC* when KNN is used. This observation suggests that some classification algorithms and datasets might *prefer* particular nonconformity functions.

***Inverse probability* is rarely the best nonconformity function**. We observed that *margin* can results 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 for a small number of cases.

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 *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 Tables 3 and 10, 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, wave, and heat datasets with RF.

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. 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*. This supports the main finding of the original paper by Johansson et al. (2017).

oneC **is not always useful.** As was demonstrated in Section 5.2.3, the 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, and the synthetic dataset with $\sigma = 0.2$. The baseline performance of the underlying classification model also has a direct impact on the efficiency of the resulting conformal classifier. 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 the results for the synthetic dataset with $\sigma = 0.4$ presented in Fig. 3. The same tendency was observed for many real-world datasets. 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* being below 1. Additionally, as it was observed for the synthetic datasets with increasing difficulties, the difference between different nonconformity functions becomes more prominent when the baseline classifiers are less accurate or the datasets are more 'difficult'. This is demonstrated by the increasing values in section 3 of Table 2. Also, we usually observe more deviations from the baseline pattern with the increase in *b_err* of the underlying classification model, see section 4 of Tables 2 and 9.

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 noncon-

formity 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 with both real-world datasets and synthetic datasets with increasing level of difficulty, 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 the KNN classifier and balance dataset. At the same time, *inverse probability* is the best choice of nonconformity function only in a small number of cases. For one synthetic dataset, we also observed the inverse relationship. However, it can be considered an exception rather than a rule, see discussion in Section 5.1.4.

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, wave and heat. 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.