Ensemble Diversity in Evolving Data Streams

Dariusz Brzezinski and Jerzy Stefanowski

Institute of Computing Science, Poznan University of Technology,
ul. Piotrowo 2, 60–965 Poznan, Poland
{dariusz.brzezinski,jerzy.stefanowski}@cs.put.poznan.pl

Abstract. While diversity of ensembles has been studied in the context of static data, it has not still received such research interest for evolving data streams. This paper aims at analyzing the impact of concept drift on diversity measures calculated for streaming ensembles. We consider six popular diversity measures and adapt their calculations to data stream requirements. A comprehensive series of experiments reveals the potential of each measure for visualizing ensemble performance over time. Measures highlighted as capable of depicting sudden and virtual drifts over time are used as basis for detecting changes with the Page-Hinkley test. Experimental results demonstrate that the κ interrater agreement, disagreement, and double fault measures, although designed to quantify diversity, provide a means of detecting changes competitive to that using classification accuracy.

Keywords: classifier ensemble, diversity measure, data stream, concept drift, drift detection

1 Introduction

Recent decades have increased interest in collecting big data, which resulted in new challenges for data storage and processing. Apart from their massive volumes, these demanding data sources are also characterized by the speed at which data is passed to analytical systems. These properties are especially relevant when data are continuously generated in the form of *data streams*.

Compared to static data, classification in streams implies new requirements for algorithms, such as constraints on memory usage, restricted processing time, and one scan of incoming examples [5,6]. An even more challenging aspect of analyzing streaming data is that learning algorithms often act in dynamic, non-stationary environments, where the data and target concepts change over time. This phenomenon, called *concept drift*, deteriorates the predictive accuracy of classifiers, as the instances the models were trained on differ from the current data. Examples of real-life concept drifts include spam categorization, weather predictions, monitoring systems, financial fraud detection, and evolving customer preferences; for their review see, e.g. [8,11,20].

Since typical batch learning algorithms for supervised classification are not capable of fulfilling the aforementioned data stream requirements, several new learning algorithms have been introduced [5,6]. They are based on using sliding windows to manage memory and provide a forgetting mechanism, sampling techniques, drift detectors, and new online algorithms. Out of several proposals, *ensemble* methods play an important role. Ensembles of classifiers are quite naturally adapted to non-stationary data streams, as they are capable of incorporating new data by either introducing a new component or updating existing components. Forgetting of outdated knowledge can be implemented by removing components that perform poorly at a given moment or by continuously adapting component weights accordingly to performance on recent data. Classifier ensembles for streaming data are typically divided into block-based (batch-incremental) and online (instance-incremental) approaches, depending on the way they process incoming examples.

Most of the existing experimental studies on stream classifiers focus on predictive abilities and computational costs of ensembles in several scenarios of concept drifts [3]. However, in earlier research on batch ensembles for static data, several researchers were also interested in the *diversity* of ensembles, which is usually calculated as the degree in which component classifiers make different decisions for a single case [12]. Some authors hypothesize that high predictive accuracy and diversity among component classifiers should be related. As a result, many researchers considered special techniques for: visualizing diversity [13], selecting the most diverse ensemble [10], or using diversity measures to prune a large pool of component classifiers [1,9,13].

On the other hand, such interest in diversity measures is not so visible in research on data stream ensembles. As ensemble components are typically learned form different parts of the data stream, potentially referring to different concept distributions, most researchers claim that they are diversified but do not measure it directly [18]. There have been rare attempts at directly promoting diversity during classifier training [14,16,19], yet once again diversity over time was not reported in these studies. Notably, Minku et al. [14] discuss the impact of diversity on online ensemble learning and reactions to drift by modifying the Poisson distribution used in Online Bagging. However, doing so they only measure accuracy of the modified ensemble, not its diversity.

In this paper, we analyze the more general problem of measuring ensemble diversity in evolving data streams. More precisely, we are interested in answering the following research questions, which are not answered by previous works:

1. Which commonly used diversity measures can be calculated for streams processed: in blocks, incrementally, incrementally with forgetting?
2. How is ensemble diversity affected by concept drifts? Does diversity change over time?
3. Do incremental component classifiers enhance or degrade diversity, compared to batch component classifiers?
4. Can diversity measures be used as additional information during classifier training or drift detection?

To answer the above questions, in the following sections we perform a review of the most popular diversity measures known from static learning, analyze the possibility of calculating these measures online, and perform a comprehensive series of experiments to evaluate the use of each measure for visualizing and detecting various types of concept-drift.

2 Related Work

2.1 Ensemble Diversity Measures

To the best of our knowledge, there have been no proposals of specialized ensemble diversity measures for changing data streams. Therefore, we will analyze the use of diversity measures known from static learning in streaming scenarios. For this purpose, we selected six popular definitions of diversity based on the comprehensive review done by Ludmila Kuncheva [12].

To illustrate the calculation of each measure, we will consider the joined outputs of two component classifiers C_i and C_j shown in Table 1. The table presents proportions of correct/incorrect answers of one of or both components, thus, the total of all the cell values $a+b+c+d=1$. An ensemble of L classifiers will produce $L(L-1)/2$ pairwise diversity values based on such tables. To get a single value we average across all pairs.

Table 1: The 2x2 ensemble component relationship table with probabilities [12]

	C_i correct	C_i wrong
C_j correct	a	b
C_j wrong	c	d

The six analyzed diversity measures are: disagreement (D), Kohavi-Wolpert variance (KW), double fault (DF), interrater agreement (κ), Yule's Q statistic (Q), and coincident failure diversity (CFD). The definitions of all the measures, using values from Table 1, are presented in Eq. 1–6. Note that we use shorter equivalents of definitions given by primary authors, to make measure descriptions shorter. For a broader discussion on ways of computing each measure, please review [12].

$$D_{i,j} = b + c \qquad (1) \qquad KW = \frac{L-1}{2L} D_{av} \qquad (2)$$

$$DF_{i,j} = d \qquad (3) \qquad \kappa = 1 - \frac{1}{2\bar{p}(1-\bar{p})} D_{av} \qquad (4)$$

$$Q_{i,j} = \frac{ad-bc}{ad+bc} \qquad (5) \qquad CFD = \begin{cases} 0, & p_0 = 1; \\ \frac{1}{1-p_0} \sum_{i=1}^{L} \frac{L-i}{L-1} p_i, & p_0 < 1. \end{cases} \qquad (6)$$

The disagreement measure D (1) is equal to the probability that two classifiers will disagree on their decision. It is worth noting that for binary classification the true label of an example is not needed to determine if components disagree. The Kovavi-Wolpert variance KW (2) is inspired by the variance of the predicted class label across different training sets that were used to build the classifier. However, here we use the property that KW differs from the averaged disagreement D_{av} by a coefficient. Double fault DF (3) counts the number of times both classifiers make mistakes, whereas κ (4) measures the level of agreement between classifiers, where $\bar{p}$ is the arithmetic mean of the components' classification accuracy. The Q statistic (5) varies between -1 and 1, where components that tend to recognize the same objects correctly will have positive values and components which tend to classify different examples incorrectly will have negative values. Finally, CFD (6) is a measure that originates from software reliability, and achieves its best value of 1 when all misclassifications are unique. In Eq. (6), p_i denotes the probability that exactly i out of L components fail on a randomly chosen input.

2.2 Stream Classifiers and Drift Detectors

As an increasingly important data mining technique, data stream classification has been widely studied by different communities; a detailed survey can be found in [5,6]. In our study, we focus on representatives of block-based and online ensembles. As an example of that first category, we will use the *Accuracy Updated Ensemble* (AUE) [3], which creates a new component with each block of examples and adds it to the ensemble, incrementally trains previously created components, and weights (evaluates) components according to their performance on the newest data block. As an example of online ensemble learning, we will use *Online Bagging* [15], which incrementally updates components with each incoming example and makes a final prediction with simple majority voting. The sampling, crucial to batch bagging, is performed incrementally by presenting each example to a component k times, where k is defined by the Poisson distribution.

In this paper, we investigate ensemble diversity measures not only as a means of visualizing ensemble and stream characteristics, but also as a basis for drift detection. For this purpose, we modify the Page-Hinkley (PH) test [7], however, generally other drift detection methods could also have been adapted [8]. The PH test considers a variable m^t, which measures the accumulated difference between observed values e (originally error estimates) and their mean till the current moment $\bar{e}^t$, decreased by a user-defined magnitude of allowed changes δ: $m^t = \sum_{i=1}^{t} (e^i - \bar{e}^t - \delta)$. After each observation e^t, the test checks whether the difference between the current m^t and the smallest value up to this moment $\min(m^i, i = 1, \ldots, t)$ is greater than a given threshold λ. If the difference exceeds λ, a drift is signaled. In this paper, we propose to use the studied diversity measures as the observed value.

3 Calculating Diversity Measures for Streaming Data

We will discuss the possibility of calculating ensemble diversity measures in three basic stream processing scenarios: in blocks, incrementally, and prequentially [6].

Block-based processing is the most natural framework for calculating diversity measures, as examples arrive in portions (chunks) of sufficient size. Thus, one can recalculate ensemble diversity on each incoming data block in a similar way as for static data. We note that each of the presented measures is based on individual component predictions and their summary in the form of pairwise component relationships presented in Table 1. Therefore, for a data block of d examples and L ensemble components, measures 1-6 can be computed in $O(d \cdot L^2)$ time and $O(d \cdot L)$ memory. Since d and L are user-defined constants, this resolves to constant time and memory per block, therefore, the analyzed measures can be successfully used in block processing.

Incremental calculation assumes that a measure can be computed based only on a summary of all previous examples and a single new example. This is slightly less trivial, however, if we monitor the number of processed examples and update a, b, c, d counts with each instance, each of the measures considered in this study can be calculated for a new example in $O(L^2)$ time and $O(L^2)$ memory.

Finally, if a stream is subject to changes, one may be interested in calculating diversity measures prequentially, that is, incrementally with forgetting [4,7]. Two basic approaches to calculating values with forgetting are used: *sliding windows* and *fading factors* [7]. Sliding windows provide a way of limiting the amount of analyzed examples by retaining a set of only d most recent examples at each time point. Fading factors, on the other hand, discount older information across time by multiplying the previous summary by a factor and adding a new value computed on the the incoming example. Sliding windows resemble data blocks updated after each example, and can be similarly used to calculate diversity measures with forgetting. Furthermore, due to the fact that all of the analyzed measures are based on counts, all of the measures can be also computed using fading factors. For this purpose, it suffices to calculate the *fading sum* $S_{x,\alpha}(t)$ and *fading increment* $N_\alpha(t)$ from a stream of objects x at time t [7]:

$$S_{x,\alpha}(t) = x^t + \alpha \times S_{x,\alpha}(t-1)$$
$$N_\alpha(t) = 1 + \alpha \times N_\alpha(t-1)$$

where x can be counts of any of the values a, b, c, d from Table 1. For example, if d^t is 1 when both components misclassify an example, then double fault can be calculated as $DF_\alpha(t) = S_{d,\alpha}(t)/N_\alpha(t)$. As with incremental computation, the prequential calculation of any of the analyzed diversity measures for a new example requires $O(L^2)$ time and $O(L^2)$ memory.

To sum up, all the considered diversity measures can be computed on blocks, incrementally, and prequentially, while fulfilling limited time and memory requirements of stream processing. As we are interested in using these diversity measures on concept-drifting data, in the following sections we will visualize and analyze diversity calculated prequentially. To the best of our knowledge, this is the first study of diversity measures from this perspective.

4 Experimental study

We performed two basic groups of experiments, one visualizing and comparing diversity measures over time, and another assessing the possibility of using them as a basis for drift detection. In the first group, we tested two different ensemble classifiers: Online Bagging (Bag) and Accuracy Updated Ensemble (AUE). Bag was chosen as an online approach, whereas AUE represents block-based ensembles. As component classifiers we compared: Naive Bayes (NB), Linear Perceptron (P), Decision trees (J48), and Hoeffding Trees (HT). For the second group of experiments, we compared drift detectors using Online Bagging with HT components.

All the algorithms and evaluation methods were implemented in Java as part of the MOA framework [2]. The experiments were conducted on a machine equipped with a dual-core Intel i7-2640M CPU, 2.8Ghz processor and 16 GB of RAM. For all the experiments, base learners where parametrized with default values proposed in MOA.

4.1 Datasets

In experiments showcasing visualizations of diversity measures over time, we used 2 real and 10 synthetic datasets[1]. For the real-world datasets it is difficult to precisely state when drifts occur. In particular, `Airlines` (`Air`) is a large, balanced dataset, which encapsulates the task of predicting whether a given flight will be delayed and no information about drifts is available. However, the second real dataset (`PAKDD`) was intentionally gathered to evaluate model robustness against performance degradation caused by market gradual changes and was studied by many research teams [17].

Additionally, we used the MOA framework [2] to generate 10 artificial datasets with different types of concept drift. The SEA generator [16] was used to create a stream without drifts ($\texttt{SEA}_{ND}$), as well as three streams with sudden changes and constant 1:1 ($\texttt{SEA}_1$), 1:10 ($\texttt{SEA}_{10}$), 1:100 ($\texttt{SEA}_{100}$) class imbalance ratios. Similarly, the Hyperplane generator [18] was used to simulate three streams with different class ratios, 1:1 ($\texttt{Hyp}_1$), 1:10 ($\texttt{Hyp}_{10}$), 1:100 ($\texttt{Hyp}_{100}$), but with a continuous incremental drift rather than sudden changes. Streams with subscripts $_{10}$ and $_{100}$ were created to assess measures in the presence of class imbalance, which usually remains undetected by classification accuracy [4]. We also tested the performance of the analyzed measures in the presence of very short, temporary changes in a stream (`RBF`) created using the RBF generator [2].

Apart from data containing real drifts, we additionally created four streams with virtual drifts, i.e., class distribution changes over time. $\texttt{SEA}_{RC}$ contains three sudden class ratio changes (1:1/1:100/1:10/1:1), whereas $\texttt{Hyp}_{RC}$ simulates a continuous ratio change from 1:1 to 1:100 throughout the stream. All the synthetic datasets, apart from `RBF`, contained 5–10% examples with class noise.

[1] Source code, test scripts, and generator parameters available at: http://www.cs.put.poznan.pl/dbrzezinski/software.php

For experiments assessing diversity measures as potential drift detectors, we created 7 synthetic datasets using the SEA (SEA), RBF (RBF), Random Tree (RT), and Agrawal (Agr) generators [2]. Each dataset tested for a single reaction (or lack of one) to a sudden change. $\texttt{SEA}_{NoDrift}$ contained no changes, and should not trigger any drift detector, while RT involved a single sudden change after 30 k examples. The $\texttt{Agr}_1$, $\texttt{Agr}_{10}$, $\texttt{Agr}_{100}$ datasets also contained a single sudden change after 30 k examples, but had a 1:1, 1:10, 1:100 class imbalance ratio, respectively. Finally, $\texttt{SEA}_{Ratio}$ included a sudden 1:1/1:100 ratio change after 10 k examples and $\texttt{RBF}_{Blips}$ contained two short temporary changes, which should not trigger the detector. The main characteristics of all the datasets are given in Table 2.

Table 2: Characteristic of datasets

Dataset	#Inst	#Attrs	Class ratio	Noise	#Drifts	Drift type
$\texttt{SEA}_{ND}$	100 k	3	1:1	10%	0	none
$\texttt{SEA}_{1}$	1 M	3	1:1	10%	3	sudden
$\texttt{SEA}_{10}$	1 M	3	1:10	10%	3	sudden
$\texttt{SEA}_{100}$	1 M	3	1:100	10%	3	sudden
$\texttt{Hyp}_{1}$	500 k	5	1:1	5%	1	incremental
$\texttt{Hyp}_{10}$	500 k	5	1:10	5%	1	incremental
$\texttt{Hyp}_{100}$	500 k	5	1:100	5%	1	incremental
RBF	1 M	20	1:1	0%	2	blips
$\texttt{SEA}_{RC}$	1 M	3	1:1/1:100/1:10/1:1	10%	3	virtual
$\texttt{Hyp}_{RC}$	500 k	3	1:1 → 1:100	5%	1	virtual
Air	539 k	7	1:1	-	-	unknown
PAKDD	50 k	30	1:4	-	-	unknown
Elec	45 k	8	1:1	-	-	unknown
KDDCup	494 k	41	1:4	-	-	unknown
$\texttt{SEA}_{NoDrift}$	20 k	3	1:1	10%	0	none
$\texttt{Agr}_{1}$	40 k	9	1:1	1%	1	sudden
$\texttt{Agr}_{10}$	40 k	9	1:10	1%	1	sudden
$\texttt{Agr}_{100}$	40 k	9	1:100	1%	1	sudden
RT	40 k	10	1:1	0%	1	sudden
$\texttt{SEA}_{Ratio}$	40 k	3	1:1/1:100	10%	1	virtual
$\texttt{RBF}_{Blips}$	40 k	20	1:1	0%	0	blips

4.2 Diversity Analysis over Time

In our first group of experiments, we plotted diversity measures (1–6) over 2 real and 10 synthetic datasets with various types of drift. The measures where prequentially calculated on a sliding window of $d = 1000$ examples for Bag and AUE. Both ensemble classifiers where tested with NB, P, J48, and HT component classifiers. For subsequent plots, we also changed the number of component classifiers $k \in \{2, 3, 4, 5, 7, 10, 15, 25, 50\}$. By changing the mentioned parameters, we are interested in assessing the influence of:

- the type of visualized diversity measure,
- type of drift occurring in the stream,
- number of component classifiers,
- type of component base learner,
- ensemble adaptation procedure.

A set of plots depicting disagreement (D) measured on Bag with different base learners and varying number of components is presented in Fig. 1. Due to the overwhelming number of subplots, we only present a full figure for D, however, a report containing plots of all the analyzed measures, for both Bag and AUE, is available online.[2]

Looking at Fig. 1, one can notice that D changes over time, and does so differently for each dataset (grid column). Moreover, subplots within one column are similar to each other. As grid rows represent the number of ensemble components, this shows that, for a given dataset, changes in diversity are not very sensitive to the ensemble size. This pattern was true for all the analyzed diversity measures.

Since the shape of each single diversity plot was very similar for varying ensemble sizes, in Fig. 2 we visually compare all the measures for Bag with fixed $k = 10$ components. The first two rows in Fig. 2 present prequentially calculated accuracy [7] and the area under the ROC curve (AUC) [4] as reference metrics, and measures CFD, D, DF, κ, KW, Q, in consecutive rows. The plot clearly showcases that the analyzed diversity measures differ from each other. For example, CFD is very sensitive to ensemble changes, whereas D, DF, and KW have relatively smooth plots. It is also worth noticing that D and KW have plots of identical shape, yet on different y-axis scales. This is expected as looking at Eq. (1) and (2), one can notice that KW is a scaled version of D. Additionally, it is worth pointing out that some of the measures seem to depict sudden (D, DF, κ) and class ratio changes (κ, CFD) over time. This suggests, that some of the analyzed measures could be monitored over time to signal drifts or problems with the performance of an ensemble.

Figure 3 presents disagreement D of Bag and AUE with $k = 10$ components on a dataset with sudden changes ($\texttt{SEA}_1$). This pair of plots shows that AUE, which periodically replaces existing components with new classifiers, showcases high variability over time. Furthermore, Fig. 3 gives a closer look at the impact of using different component base learners. The Naive Bayes (NB) algorithm does not promote diversity among components and does not depict diversity changes over time. The Hoeffding Tree (HT) and Perceptron (P) are much better at depicting changes over time due to their incremental nature. Finally, batch decision trees (J48) are only applicable to block-based ensembles. These properties were shared by all the analyzed plots.

In the following section, we will take a closer look at the possibility of detecting drifts using ensemble diversity measures.

[2] http://www.cs.put.poznan.pl/dbrzezinski/software/DiversityInStream.html

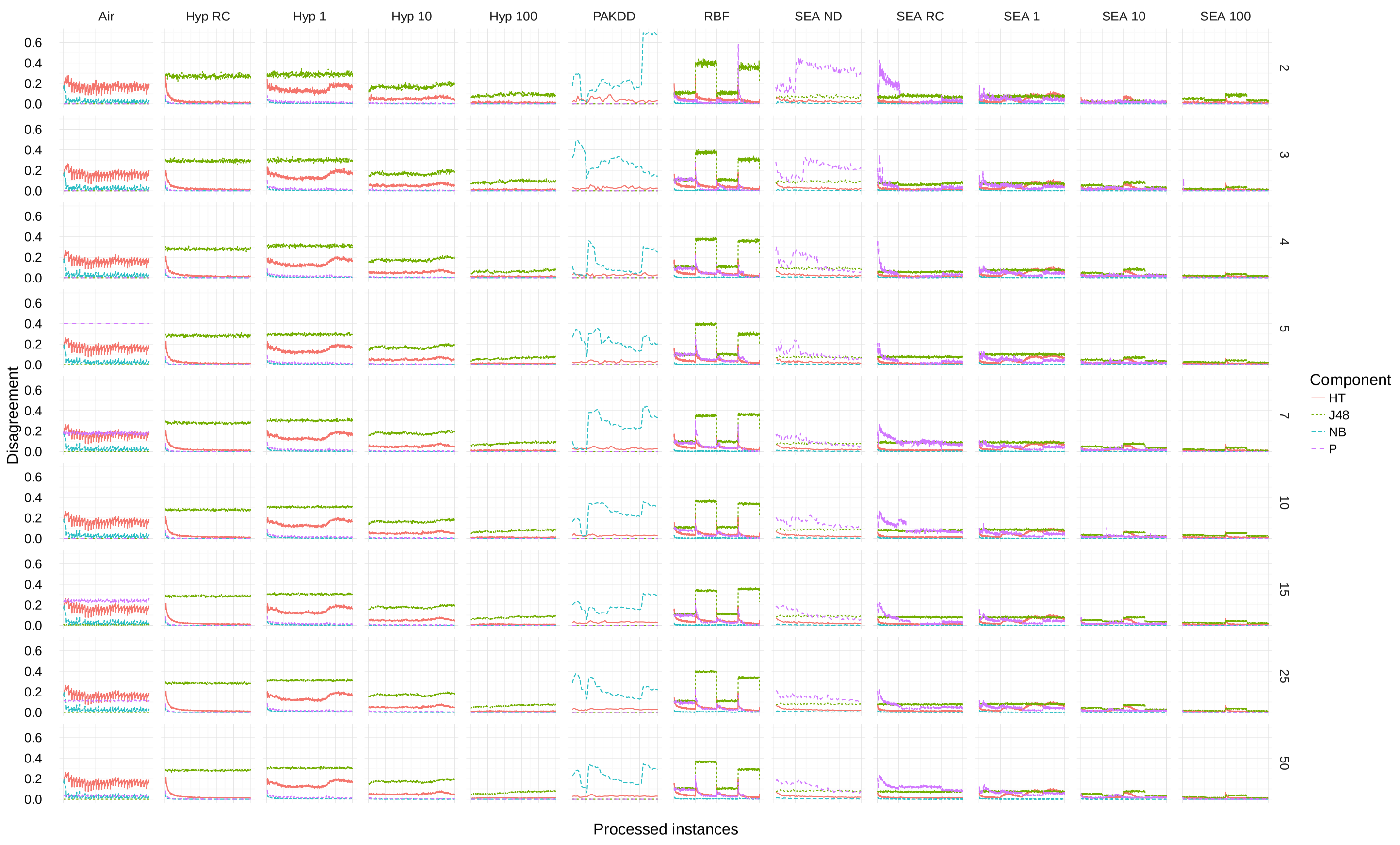


Fig. 1: Disagreement visualizations on all the analyzed datasets for Bag with $k \in \{2, 3, 4, 5, 7, 10, 15, 25, 50\}$ components

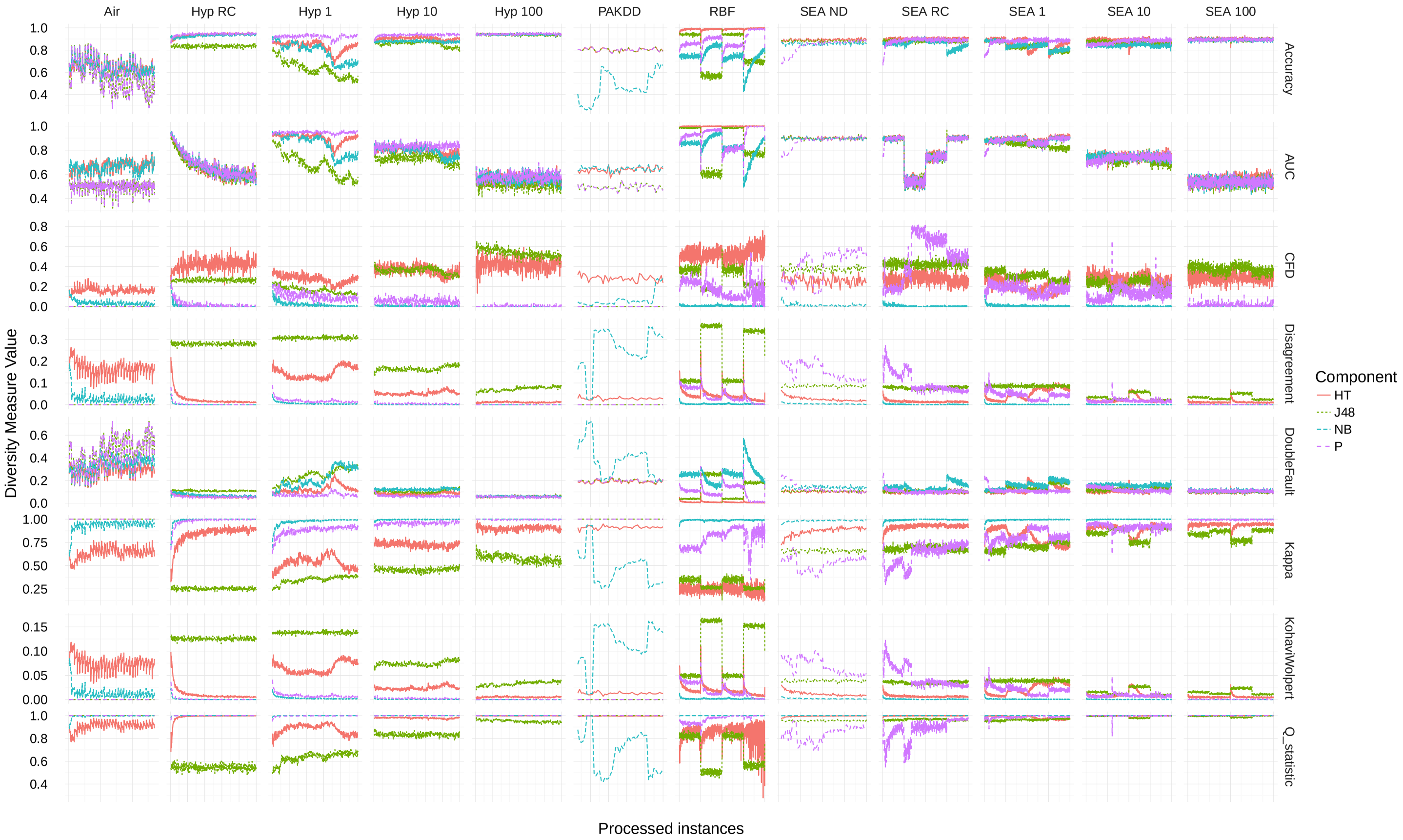
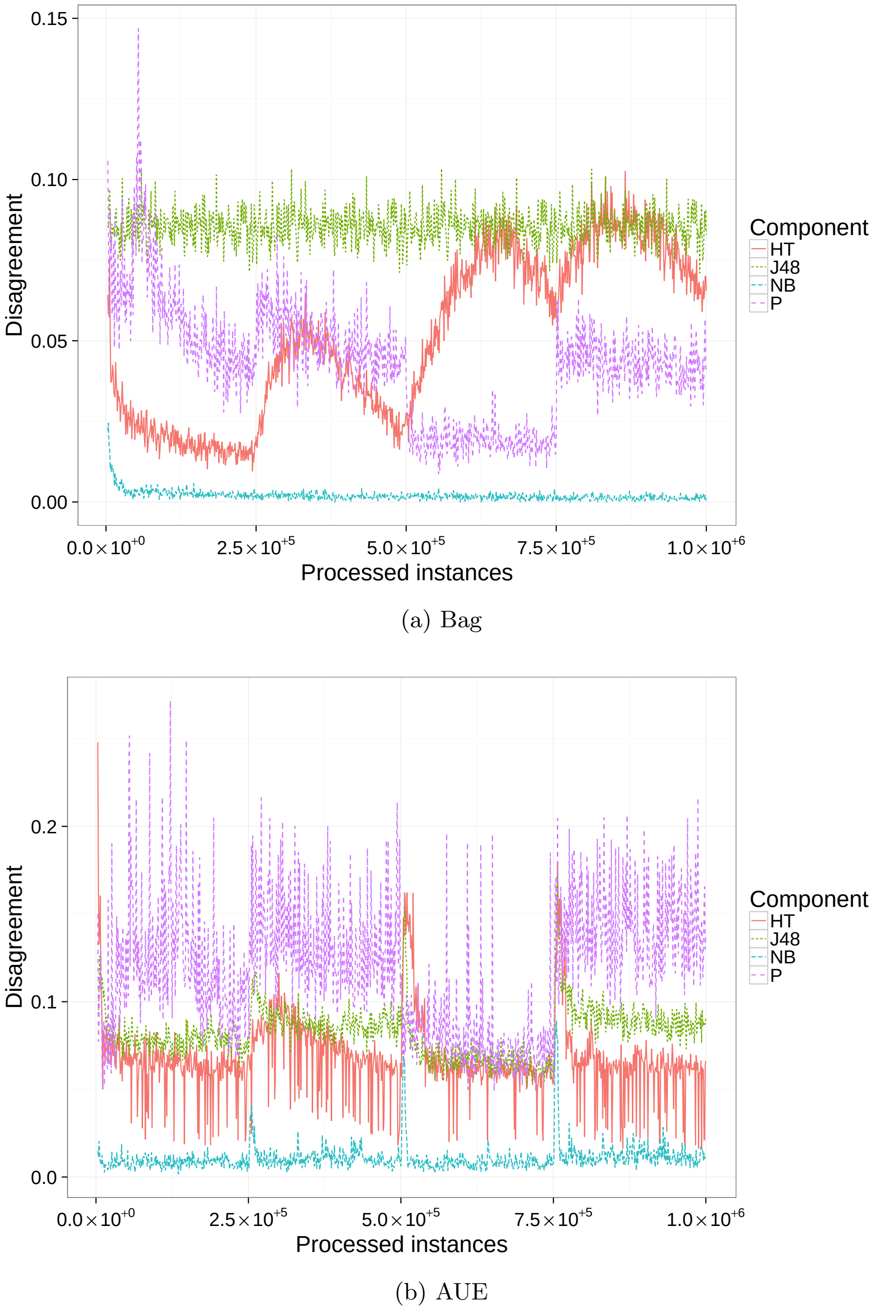


Fig. 2: Performance and diversity measure visualizations on all the analyzed datasets for Bag with $k = 10$ components



(a) Bag

(b) AUE

Fig. 3: Comparison of disagreement D visualizations of Bag and AUE with $k = 10$ components for a stream with sudden changes ($\mathtt{SEA}_1$)

4.3 Drift Detection using Diversity Measures

The second group of experiments involved using the PH test to detect drifts based on changes in prequential accuracy and diversity measures. To compare all the analyzed metrics, we used sliding window sizes (1000–5000) and PH test parameters ($\lambda = 100$, $\delta = 0.1$) as proposed in [7]. Table 3 presents the number of missed versus false detection counts, with average delay time for correct detections; subscripts in column names indicate the PH test window size. The results refer to total counts and means over 10 runs of streams generated with different random seeds.

First, we note that two diversity measures, Q and KW, are missing from Table 3. We omitted these two measures from the presentation, because the drift detector never triggered for these measures. That means that for datasets with drifts Q and KW always had 10 missed detections and 0 false alarms. Thus, our first observation is that Q and KW are not good candidates for drift monitors, at least when using the PH Test. One explanation of this fact may be that the Kohavi-Wolpert variance KW is a measure with a small range of values, which most probably makes it difficult for the detector to trigger. The Q statistic, on the other hand, puts common misclassifications and correct answers on two ends of its scale, this way introducing difficulties for the used PH Test.

Another outlying measure is CFD. The Coincident Failure Diversity is very susceptible to small changes in the ensemble, causing a very large number of false alarms. Therefore, just as Q and KW, CFD is not a good choice of monitored value when detecting drifts using the PH Test.

The remaining three measures (κ, DF, D) showcase good drift detection properties. Particularly, κ offers detection rates comparable to those of prequential accuracy with smaller delay. Additionally, κ successfully detected 9 out of 10 class ratio changes, whereas accuracy did not detect any of them. DF and D have slightly more missed detections and are slower at signaling changes. However, it is worth noting that for binary classification problems, D has the potential of working in unlabeled or partially labeled stream settings. This opens an interesting option for future research, and might mean that if predictions of components start to disagree in an unusual way, we may be able to observe sudden changes even without true labels of incoming examples.

5 Conclusions and Outlook

Diversity is often perceived as one of the most important characteristics of ensemble classifiers. However, even though ensembles are among the most often proposed approaches for concept-drifting streams, up till now ensemble diversity measures have not been thoroughly studied in the context of time-evolving data. In this paper, we reviewed diversity measures known from static data, and analyzed the possibility of calculating them on blocks, incrementally, and prequentially. Additionally, a comprehensive series of experiments was performed to evaluate the use of each measure for visualizing and detecting various types of concept-drift.

Table 3: Number of missed and false detections (in the format missed:false) obtained using the PH test with prequential accuracy and diversity measures. Mean delays of correct detections are given in parenthesis, where (-) means that the detector was not triggered or the dataset did not contain any change.

	Acc$_{1k}$	Acc$_{2k}$	Acc$_{3k}$	Acc$_{4k}$	Acc$_{5k}$
SEA$_{NoDrift}$	0:0 (-)	0:0 (-)	0:0 (-)	0:0 (-)	0:0 (-)
Agr$_1$	0:0 (946)	0:0 (1614)	0:0 (2265)	0:0 (2920)	0:0 (3582)
Agr$_{10}$	0:5 (805)	1:6 (1287)	0:1 (1685)	0:1 (2197)	0:1 (2909)
Agr$_{100}$	4:13 (1416)	4:11 (1706)	5:13 (2637)	4:10 (3035)	4:9 (3748)
RT	6:0 (1851)	7:0 (2414)	7:0 (3428)	8:0 (3656)	8:0 (4514)
SEA$_{Ratio}$	10:0 (-)	10:0 (-)	10:0 (-)	10:0 (-)	10:0 (-)
RBF$_{Blips}$	0:0 (-)	0:0 (-)	0:0 (-)	0:0 (-)	0:0 (-)
	κ_{1k}	κ_{2k}	κ_{3k}	κ_{4k}	κ_{5k}
SEA$_{NoDrift}$	0:0 (-)	0:0 (-)	0:0 (-)	0:0 (-)	0:0 (-)
Agr$_1$	0:0 (608)	0:0 (986)	0:0 (1352)	0:0 (1719)	0:0 (2082)
Agr$_{10}$	0:6 (453)	1:8 (648)	5:8 (757)	1:7 (1115)	1:8 (1810)
Agr$_{100}$	10:10 (-)	8:3 (596)	9:3 (1945)	9:3 (2769)	9:1 (3558)
RT	5:0 (1456)	6:0 (2057)	6:0 (2890)	6:0 (3809)	6:0 (4851)
SEA$_{Ratio}$	1:0 (1073)	1:0 (1976)	1:0 (2874)	1:0 (3755)	1:0 (4635)
RBF$_{Blips}$	0:0 (-)	0:0 (-)	0:0 (-)	0:0 (-)	0:0 (-)
	DF_{1k}	DF_{2k}	DF_{3k}	DF_{4k}	DF_{5k}
SEA$_{NoDrift}$	0:0 (-)	0:0 (-)	0:0 (-)	0:0 (-)	0:0 (-)
Agr$_1$	0:0 (1200)	0:0 (2112)	0:0 (3051)	0:0 (4019)	0:0 (5027)
Agr$_{10}$	0:3 (881)	0:1 (1387)	0:1 (1938)	0:1 (2727)	0:1 (3817)
Agr$_{100}$	10:10 (-)	10:5 (-)	10:5 (-)	10:5 (-)	10:5 (-)
RT	6:0 (2125)	8:0 (2092)	8:0 (2881)	8:0 (3688)	8:0 (4561)
SEA$_{Ratio}$	10:0 (-)	10:0 (-)	10:0 (-)	10:0 (-)	10:0 (-)
RBF$_{Blips}$	0:0 (-)	0:0 (-)	0:0 (-)	0:0 (-)	0:0 (-)
	D_{1k}	D_{2k}	D_{3k}	D_{4k}	D_{5k}
SEA$_{NoDrift}$	0:0 (-)	0:0 (-)	0:0 (-)	0:0 (-)	0:0 (-)
Agr$_1$	1:0 (1582)	1:0 (2342)	1:0 (3154)	1:0 (4008)	1:0 (4909)
Agr$_{10}$	9:1 (3120)	9:1 (3885)	9:0 (4704)	9:0 (5580)	9:0 (6464)
Agr$_{100}$	7:12 (3693)	8:8 (2337)	8:4 (4818)	9:3 (2991)	9:3 (4272)
RT	10:0 (-)	10:0 (-)	10:0 (-)	10:0 (-)	10:0 (-)
SEA$_{Ratio}$	10:0 (-)	10:0 (-)	10:0 (-)	10:0 (-)	10:0 (-)
RBF$_{Blips}$	0:0 (-)	0:0 (-)	0:0 (-)	0:0 (-)	0:0 (-)
	CFD_{1k}	CFD_{2k}	CFD_{3k}	CFD_{4k}	CFD_{5k}
SEA$_{NoDrift}$	0:176 (-)	0:120 (-)	0:80 (-)	0:59 (-)	0:41 (-)
Agr$_1$	0:580 (281)	0:284 (558)	0:176 (1449)	0:121 (1903)	0:93 (1386)
Agr$_{10}$	0:516 (506)	0:267 (908)	0:181 (741)	0:131 (1919)	0:102 (1673)
Agr$_{100}$	0:190 (1242)	0:187 (1085)	0:148 (1294)	0:113 (2463)	0:95 (2073)
RT	0:676 (218)	0:330 (531)	0:213 (810)	0:152 (1117)	0:117 (1972)
SEA$_{Ratio}$	1:218 (2501)	0:158 (1483)	0:114 (3152)	0:100 (2402)	0:80 (2067)
RBF$_{Blips}$	0:533 (-)	0:260 (-)	0:170 (-)	0:120 (-)	0:100 (-)

Regarding the first question posed in the introduction, we can state that all six of the analyzed measures can be adapted to data stream requirements and computed according to three basic processing paradigms: on consecutive blocks, incrementally, and prequentially. We find the answer to the second question much more interesting. Visualizations of diversity measures calculated on streams with various types of drifts have shown that ensemble diversity visibly changes over time. In particular, we were able to highlight κ interrater agreement, double fault, disagreement, and coincident failure diversity, as measures that were able to depict sudden changes. Additionally, it is worth noting that diversity of the tested ensembles was generally low in terms of absolute values, which might signal that there is still pending research in the field of adaptive ensembles.

The third research question raised the problem of using incremental versus batch classifiers as ensemble components. Our results show that incremental base learners have greater potential for depicting diversity over time. In particular, Hoeffding trees and linear perceptrons were better at visualizing changes over time than batch decision trees and the Naive Bayes algorithm. We also noticed differences between using an adaptive ensemble that only updates existing components and one that periodically creates new components. The latter, represented in our experiment by the Accuracy Updated Ensemble exhibits slightly higher but also much more variable diversity. Surprisingly, one of the most commonly tuned ensemble parameters, the number of components, showcased little impact on diversity plots over time.

Finally, we were interested whether diversity measures can be used to detect concept drifts. A separate set of experiments employing the Page-Hinkley test showed that κ interrater agreement, double fault, and disagreement are capable of detecting sudden changes. In particular, κ was capable of detecting changes equally effectively as accuracy, with smaller delays. Moreover, contrary to accuracy, κ was capable of detecting class ratio changes.

Observations made in this study open several lines of future research. Drawing parallels from static ensembles, diversity measures could be used to prune large ensembles online. Moreover, the complementary nature of various diversity and performance measures suggests it might be worth investigating ideas of combining multiple detectors, which would monitor more than one metric. It is also worth recalling that for binary classification the true label of an example is not needed to calculate disagreement. Thus, there might be a possibility of using disagreement to detect sudden drifts in partially labeled streams, where supervised detectors cannot be applied. Finally, data stream characteristics call for new specialized diversity measures and visualizations. For example, one could take into account differences in component age when calculating pairwise diversity measures or visualize the variability of disagreement among component pairs by using whiskers or box plots.

Acknowledgments. The authors' research was funded by the Polish National Science Center under Grant No. DEC-2013/11/B/ST6/00963. Dariusz Brzezinski acknowledges the support of an FNP START scholarship.

References

1. Banfield, R.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P.: A new ensemble diversity measure applied to thinning ensembles. In: Proc. 4th Int. Workshop Multiple Classifer Systems. LNCS, vol. 2709, pp. 306–316 (2003)
2. Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B.: MOA: Massive Online Analysis. J. Mach. Learn. Res. 11, 1601–1604 (2010)
3. Brzezinski, D., Stefanowski, J.: Reacting to different types of concept drift: The accuracy updated ensemble algorithm. IEEE Trans. on Neural Netw. Learn. Syst. 25(1), 81–94 (2014)
4. Brzezinski, D., Stefanowski, J.: Prequential AUC for classifier evaluation and drift detection in evolving data streams. In: New Frontiers in Mining Complex Patterns. Lecture Notes in Computer Science, vol. 8983, pp. 87–101 (2015)
5. Ditzler, G., Roveri, M., Alippi, C., Polikar, R.: Learning in nonstationary environments: A survey. IEEE Comp. Intell. Mag. 10(4), 12–25 (2015)
6. Gama, J.: Knowledge Discovery from Data Streams. Chapman and Hall (2010)
7. Gama, J., Sebastião, R., Rodrigues, P.P.: On evaluating stream learning algorithms. Mach. Learn. 90(3), 317–346 (2013)
8. Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM Comput. Surv. 46(4), 44:1–44:37 (2014)
9. Giacinto, G., Roli, F.: An approach to the automatic design of multiple classifier systems. Pattern Recognition Letters 22(1), 25–33 (2001)
10. Giacinto, G., Roli, F.: Design of effective neural network ensembles for image classification purposes. Image Vision Comput. 19(9-10), 699–707 (2001)
11. Krempl, G., Zliobaite, I., Brzezinski, D., Hüllermeier, E., Last, M., Lemaire, V., Noack, T., Shaker, A., Sievi, S., Spiliopoulou, M., Stefanowski, J.: Open challenges for data stream mining research. SIGKDD Explorations 16(1), 1–10 (2014)
12. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)
13. Margineantu, D.D., Dietterich, T.G.: Pruning adaptive boosting. In: Proc. 14th Int. Conf. Mach. Learn. pp. 211–218 (1997)
14. Minku, L.L., White, A.P., Yao, X.: The impact of diversity on online ensemble learning in the presence of concept drift. IEEE Trans. Knowl. Data Eng. 22(5), 730–742 (2010)
15. Oza, N.C., Russell, S.J.: Experimental comparisons of online and batch versions of bagging and boosting. In: Proc. 7th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining. pp. 359–364 (2001)
16. Street, W.N., Kim, Y.: A streaming ensemble algorithm (SEA) for large-scale classification. In: Proc. 7th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining. pp. 377–382 (2001)
17. Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.B.: PAKDD data mining competition (2009)
18. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: Proc. 9th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining. pp. 226–235 (2003)
19. Wozniak, M.: Application of combined classifiers to data stream classification. In: Proc. 12th Int. Conf. Computer Information Systems and Industrial Management. LNCS, vol. 8104, pp. 13–23 (2013)
20. Zliobaite, I., Pechenizkiy, M., Gama, J.: An overview of concept drift applications. In: Japkowicz, N., Stefanowski, J. (eds.) Big Data Analysis: New Algorithms for a New Society, Studies in Big Data, vol. 16, pp. 91–114. Springer (2016)