Cosine Similarity Drift Detector

Juan I. G. Hidalgo, Laura M. P. Mariño and Roberto Souto Maior de Barros
Centro de Informática, Universidade Federal de Pernambuco, Brazil
{jigh, lmpm, roberto}@cin.ufpe.br

Introduction

- In the information age, large amounts of data are constantly generated over time, which are known as data streams. One of the difficulties in dealing with streaming data is the fact that the concepts (data distribution) can change over time.
- These changes in the distribution of the problem are known as concept drifts. The speed with which such changes occur may be categorized as:
  - Abrupt: when the transition from an old to a new concept occurs suddenly, or
  - Gradual: when such a transition is smooth.
- In the process of learning from data streams with concept drift, the Cosine similarity measure has previously proved to be an excellent metric in assessments with imbalanced datasets.
- Our main motivation is the fact that this measure had not yet been evaluated as a tool to compare the similarity between two data distributions using sliding windows to detect concept drifts.
- This paper proposes CSDD, a new method that compares recent and older data using the Cosine similarity and windowing techniques aiming to detect concept drifts.
- To validate it, experiments were run using both synthetic and real-world datasets, and Naive Bayes (NB) and Hoeffding Tree (HT) as base learners.
- The experimental results show the effectiveness of CSDD in scenarios with abrupt and gradual changes as it delivered the best results in nearly all artificial datasets.

Cosine Similarity Drift Detector

- CSDD works very similarly to the Wilcoxon Rank Sum Test Drift Detector (WSTD):
  - They monitor predictions of the base classifier using two windows, named as recent and old, and their sizes in instances are w and w_2, respectively.
  - The number of examples of the older window of data is also limited, as in WSTD, instead of the unlimited older window adopted by STEPD.

Algorithm 2: Cosine Similarity Drift Detector

Input: Data stream s, Recent window size w, Drift level α_d, Warning level α_w, Older window size w_2

1 $storedPreds \leftarrow$ **new byte** $[w]$
2 $storedPreds_2 \leftarrow$ **new byte** $[w_2]$
3 $vector_A \leftarrow$ **new double** $[2]$
4 $vector_B \leftarrow$ **new double** $[2]$
5 $n_o \leftarrow n_r \leftarrow w_o \leftarrow w_r \leftarrow r_o \leftarrow r_r \leftarrow 0$
6 $changeDetected \leftarrow$ **false**
7 **foreach** *instance* **in** s **do**
8 **if** $changeDetected$ **then**
9 **reset** $storedPreds, storedPreds_2$
10 $n_o \leftarrow n_r \leftarrow w_o \leftarrow w_r \leftarrow r_o \leftarrow r_r \leftarrow 0$
11 $changeDetected \leftarrow$ **false**
12 Updates predictions in *older* and *recent* windows
13 Updates stats of both windows: $n_o, n_r, w_o, w_r, r_o, r_r$
14 $isWarningZone \leftarrow$ **false**
15 **if** $n_o \geq w$ **then**
16 $rateppv_o \leftarrow r_o \,/\, (r_o + w_o)$
17 $ratefdr_o \leftarrow w_o \,/\, (w_o + r_o)$
18 $rateppv_r \leftarrow r_r \,/\, (r_r + w_r)$
19 $ratefdr_r \leftarrow w_r \,/\, (w_r + r_r)$
20 $vector_A \leftarrow [rateppv_o, ratefdr_o]$
21 $vector_B \leftarrow [rateppv_r, ratefdr_r]$
22 $sp \leftarrow$ **scalarProduct** $(|vector_A, vector_B|)$
23 $sqva \leftarrow$ **squareVector** $(|vector_A|)$
24 $sqvb \leftarrow$ **squareVector** $(|vector_B|)$
25 $S \leftarrow sp \,/\, (\mathbf{sqrt}\,(sqva) \times \mathbf{sqrt}\,(sqvb))$
26 **if** $S < \alpha_d$ **then**
27 $changeDetected \leftarrow$ **true**
28 **else if** $S < \alpha_w$ **then**
29 $isWarningZone \leftarrow$ **true**

- CSDD receives as inputs a data stream, the sizes of the two windows, and the drift and warning levels.
- Two vectors *A* and *B* are used in the calculation of the Cosine similarity of the data in the two windows.
- After a concept drift is detected, the necessary adjustments in the two windows, two vectors and the other local variables are implemented (lines 8-11).
- Whenever a new instance of the data stream is processed, the required updates and statistics are made to the windows (lines 12-13).
- The calculations and detections of drifts and warnings only occur after the older window is least the same size as the recent window, i.e. w instances (line 15).
- The computation of the *PPV* and *FDR* rates of the two windows are associated to the values of the correct (r_o, r_r) and wrong (w_o, w_r) predictions of the classifier, to determine the rate calculations that are quantified in both windows. To permit the calculations, we associated r_o and r_r to *TP* whereas w_o and w_r were associated to *FP* (lines 16-19).
- After the computed rates are stored in the vectors, the calculation of the Cosine similarity between vectors A and B is implemented (lines 22-25).
- Finally the tests used to decide the position of drifts and warnings are represented (lines 26-29).

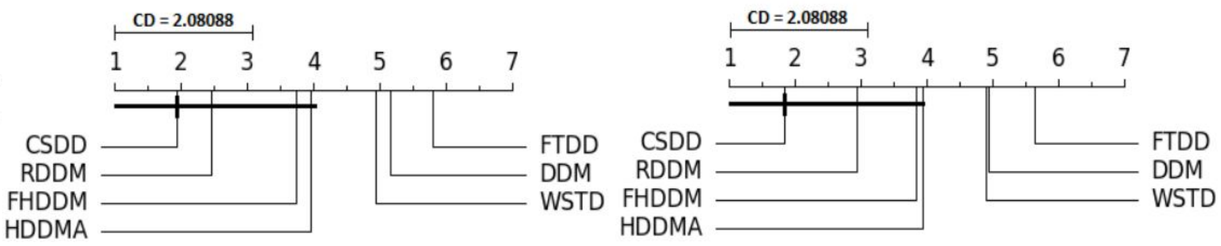
Analysis: Accuracy and Concept Drift Identifications

TABLE I: Mean accuracies in percentage using HT with 95% confidence intervals in scenarios of abrupt and gradual concept drifts with artificial and real datasets.

Type	Dataset	DDM	FHDDM	FTDD	HDDMA	RDDM	WSTD	CSDD
Abr	Agraw	63.13±0.59	64.48 ±0.40	62.64±0.40	64.47±0.36	64.69±0.32	63.44±0.45	**65.58±0.44**
	LED	69.56±0.31	69.29±0.77	67.01±0.39	69.68±0.31	69.78±0.31	67.08±1.05	**69.98±0.32**
	Mixed	89.70±0.30	90.14±0.23	90.33±0.22	90.32±0.24	90.17±0.25	90.36±0.23	**90.39±0.24**
	Sine	87.01±0.76	88.29±0.17	88.37±0.16	88.39±0.17	87.98±0.21	88.38±0.15	**88.44±0.21**
	Wavef	78.45±0.48	78.99 ±0.61	78.07±0.47	78.69±0.51	79.09±0.49	78.77±0.53	**79.38±0.47**
Grad	Agraw	61.57±0.49	62.56±0.31	61.33±0.25	62.27±0.38	62.92±0.28	61.77±0.40	**63.18±0.22**
	LED	67.76±0.44	66.86±0.93	62.88±0.37	67.58±0.32	**67.81±0.30**	63.99±0.84	67.45±0.31
	Mixed	83.49±0.29	83.98±0.24	83.50±0.29	83.39±0.28	83.70±0.32	83.26±0.29	**84.25±0.27**
	Sine	82.43±0.30	82.95±0.25	82.28±0.22	82.41±0.28	82.66±0.20	82.14±0.23	**83.21±0.21**
	Wavef	77.97±0.45	77.77 ±0.45	76.68±0.41	77.82±0.49	78.42±0.38	77.57±0.53	**78.66±0.39**
Real	Airlines	65.30	65.37	64.75	65.00	**66.01**	65.15	65.87
	KDD	97.78	97.71	97.71	**97.85**	97.36	97.71	89.73
	Rialto	36.88	42.73	30.83	45.70	**48.17**	37.38	43.07
	Usenet2	68.29	68.48	**69.10**	68.61	68.58	68.48	**69.10**
	WhiteWine	43.33	46.09	43.32	43.31	43.84	45.27	**46.11**
Rank	Artificial	5.0000	3.9000	5.9000	4.0000	2.8000	5.1000	**1.3000**
	Real	4.8000	3.7000	5.1000	3.8000	3.2000	4.5000	**2.9000**
	All	4.9333	3.8333	5.6333	3.9333	2.9333	4.9000	**1.8333**

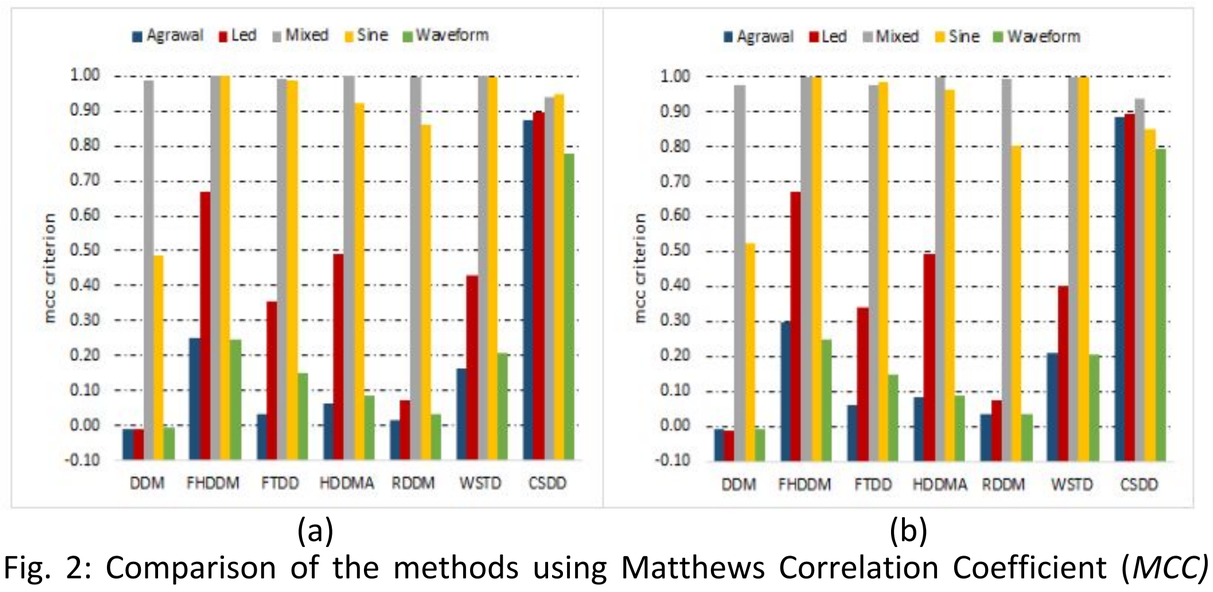
TABLE II: Mean concept drift identifications of the methods in the abrupt datasets using NB and HT as base classifiers.

	NB							HT					
Detect.	μD	FN	FP	Prec	Rec	MCC	DS	μD	FN	FP	Prec	Rec	MCC
DDM	27.3	83.0	65.8	0.284	0.308	0.289		27.8	81.0	70.2	0.283	0.325	0.295
FHDDM	19.0	45.2	43.0	0.660	0.623	0.634	**M**	19.3	43.6	42.6	0.667	0.637	0.643
FTDD	22.1	61.4	28.0	0.539	0.488	0.504	**E**	22.1	60.8	35.0	0.530	0.493	0.503
HDDMA	20.9	60.0	40.8	0.533	0.500	0.514	**A**	19.6	58.8	38.8	0.547	0.510	0.525
RDDM	26.9	70.4	63.4	0.384	0.413	0.396	**N**	24.6	69.4	74.4	0.365	0.422	0.388
WSTD	20.4	51.6	62.6	0.572	0.570	0.559		20.2	50.6	71.0	0.573	0.578	0.563
CSDD	**5.2**	**7.0**	**22.4**	**0.839**	**0.942**	**0.887**		**5.5**	**7.8**	**25.6**	**0.816**	**0.935**	**0.872**



(a) Naive Bayes (b) Hoeffding Tree

Fig. 1: Accuracy statistical comparison of the methods using the Friedman test and the Bonferroni-Dunn post-hoc test on all tested datasets.



(a) (b)

Fig. 2: Comparison of the methods using Matthews Correlation Coefficient (*MCC)* criterion on abrupt datasets with Naive Bayes (*a*) and Hoeffding Tree (*b*).

Conclusion and Future Work

- The results of the experiments suggests the superiority of CSDD against the other tested methods on both abrupt and gradual datasets in terms of accuracy as well as of the detections of the drifts. However, in the real-world datasets the results were not as strong, especially using NB as base learner.
- The results of the experiments were also evaluated statistically using a variation of the *Friedman* test in combination with the *Bonferroni-Dunn* post-hoc test.
  - This evaluation confirmed the superiority of CSDD to the other tested approaches as it was ranked first in the tests performed with both base learners, despite no statistical difference to RDDM, FHDDM and HDDMA.
- Similar findings were obtained in the evaluation using with the Matthews Correlation Coefficient (MCC) criterion which confirmed that CSDD was also the best method in the detections of concept drifts.
- We claim that CSDD surpassed the current state of art detectors, showing that the Cosine similarity together with sliding windows and *PPV* and *FDR* rates can also produce excellent results in the detection of concept drifts.
- As future work, we plan to make further experimentation and evaluations of the proposed method in order to improve its performance and make it more competitive in the real-world datasets. Finally, CSDD should also be evaluated using imbalanced datasets.