Covariate Shift Detection Based on Exponentially Weighted Moving Average

Yuhua Li

Data Analytics and Machine Learning
Cardiff University
Email: LiY180@Cardiff.ac.uk

Presented online: ONS-UNECE ML Group

21st September 2022

Outline

- Introduction to dataset shift/drift
- Shift-Detection based on Exponentially Weighted Moving Average (**SD-EWMA**)
- Two-stage Dataset Shift-detection Based on EWMA (**TSSD-EWMA**)
- Adaptive Learning with Covariate Shift–Detection (**ALCSD**)
- Conclusion

Introduction

- Classical **learning systems** are built upon the assumption that the **input data distribution for the training** and **testing** are the same.

- Real-world environments are often **non-stationary**
  - e.g., spending behaviour before and after the cost of living (energy) crisis

- So, **learning** in real-time environments is **difficult** due to the **non-stationarity effects** and the performance of the system **degrades** with time.

- This research problem can be addressed by:

  - **Non-stationary shift-detection test**

Supervised Learning

- Training samples: Input (x) and output (y)
- Learn input-output rule: $y = \hat{f}(x)$

- Assumption: "**Training** and **test** samples are drawn from **same probability distribution**" i.e., $(P_{train}(y, x) = P_{test}(y, x))$

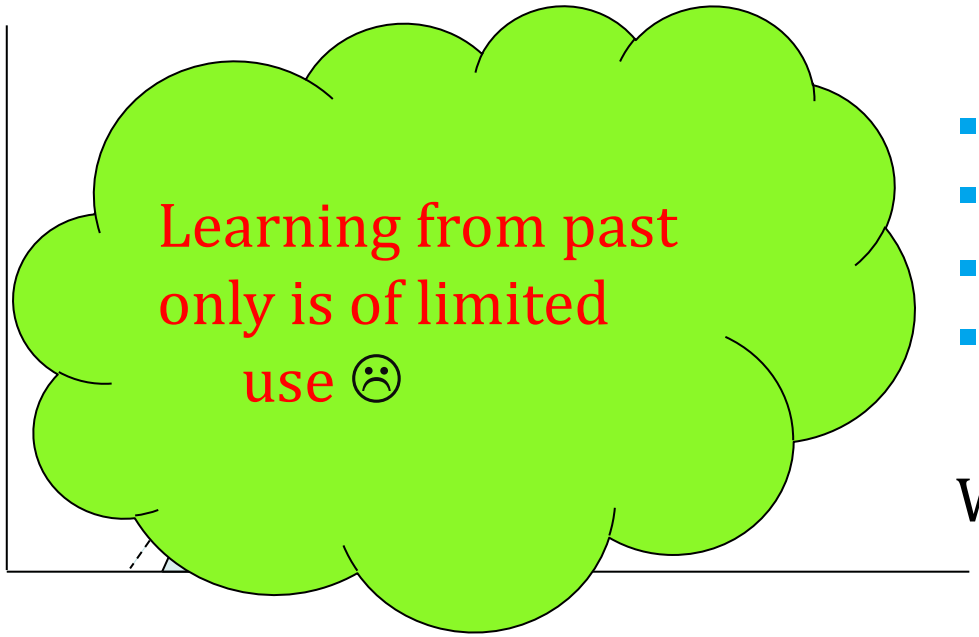
Is this assumption is really true?

Non-Stationarity

Definition: Let $(X_t)_{t \in I}$ be a multivariate time-series, where $I \subset \mathbb{R}$ is an index set. Then (X_t) is called **stationary time-series**, if the probability distribution does not change over time, i.e.,

$$P\left(X_{t_{(i)}}\right) = P\left(X_{t_{(j)}}\right)$$

for all $t_{(i)}, t_{(j)} \in I$. A time-series is called **non-stationary**, if it is not stationary.



For examples:

- Brain-computer interface
- Robot control
- Remote sensing application
- Network intrusion detection

What is the challenge?

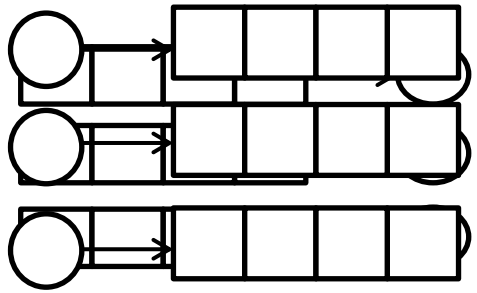
Dataset Shift

Dataset Shift appears when **training** and **test** joint distributions are **different**. That is, when $P_{train}(y, x) \neq P_{test}(y, x)$

*__Note__ : Relationship between **covariates (x)** and **class label (y)**

X→Y: Predictive model (e.g., spam filtering)

Y→X: Generative model (e.g., medical prognosis)



Types of Dataset Shift

- Covariate Shift
- Prior Probability Shift
- Concept Shift

Concept shifts appears

$$p_{train}(y|x) \neq p_{test}(y|x), \&$$
$$p_{train}(x) = p_{test}(x) in$$
$$X \rightarrow Y$$

Torres, et. al. (2012) “A unifying view on dataset shift in classification,” Pattern Recognition 45(1), 521–530.

Dataset Shift-Detection

Detecting **abrupt** and **gradual** shifts in time-series data is called **dataset shift detection**.

Types of Shift-Detection

- **Retrospective/offline-detection:** (i.e., Shift-point analysis)
- **Real-time/online-detection:** (i.e., Control charts)

The Proposed dataset shift detection

- Shift-Detection based on Exponentially Weighted Moving Average **(SD-EWMA)**
- Two-stage Dataset Shift-detection Based on EWMA **(TSSD-EWMA)**

Shift-Detection based on Exponentially Weighted Moving Average (SD-EWMA)

$$z_{(i)} = \lambda . x_{(i)} + (1 - \lambda) \cdot z_{(i-1)} \quad (1)$$

where **λ** is the **smoothing constant** (0<λ≤1).

It is a first-order autoegressive integrated moving average (ARIMA) model.

$$x_{(i)} = x_{(i-1)} + \varepsilon_i - \theta \cdot \varepsilon_{i-1} \quad (2)$$

where ε_i is a sequence of i.i.d random signal with zero mean and constant variance.

Equation (1) with $\lambda = 1 - \theta$, is the optimal **1-step-ahead prediction** for this process $\hat{x}_{(i+1)}(i) = z_{(i)}$

The **1-step-ahead** error are calculated as

$$err_{(i)} = x_{(i)} - z_{(i-1)} \quad (3)$$

If the **1-step-ahead error** is normally distributed, then

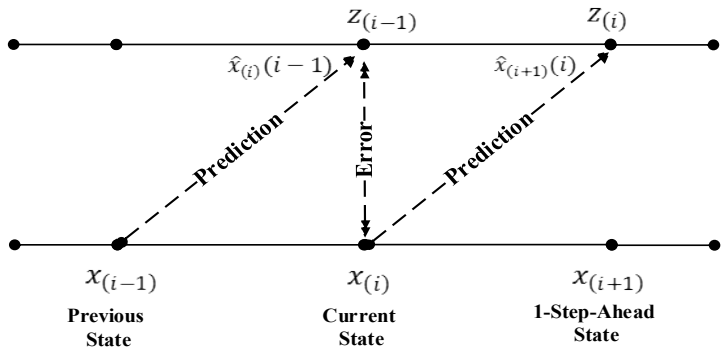
$$P\left[-L \cdot \sigma_{err} \leq err_{(i)} \leq L \cdot \sigma_{err}\right] = 1 - \alpha$$

$$P\left[-L.\,\sigma_{err} \leq x_{(i)} - z_{(i-1)}\,(i) \leq L.\,\sigma_{err}\right] = 1 - \alpha$$

LCL

UCL

$$P\left[z_{(i-1)} - L.\,\sigma_{err} \leq x_{(i)} \leq z_{(i-1)} + L.\,\sigma_{err}\right] = 1 - \alpha$$



Algorithm SD-EWMA

Input: Submit the training dataset to the training phase and compute the parameters for testing.
Receive new data in testing phase sample-by-sample and perform the check as follows.
IF (Shift detected)
THEN (Report the point of shift and initiate an appropriate corrective action)
ELSE (Continue and integrate the upcoming information).
Output: Shift-detection points

Training Phase

1. Assign training data to $x_{(i)}\ for\ i = 1\ to\ n$, n is the size of training data.
2. Calculate the mean of input data $(\bar{x})$ and assign it to $z_{(0)}$.
3. Compute the z-statistics for each observation $x_{(i)}$in training data for a range of λ values.
$$z_{(i)} = \lambda . x_{(i)} + (1 - \lambda) . z_{(i-1)}$$
4. Compute 1-step-ahead prediction errors
$$err_{(i)} = x_{(i)} - z_{(i-1)}$$
5. Estimate λ by minimizing the sum of the squared prediction error on the training data.
6. Finally compute the sum of the square of 1-step-ahead prediction error divided by the number of observations and use it as the initial value of the variance $(\sigma_{err}{}^{2}_{(0)})$ for the testing phase.

Testing Phase

1. For each data point $x_{(i)}$ in the operation/testing phase,
2. Compute $z_{(i)} = \lambda . x_{(i)} + (1 - \lambda) . z_{(i-1)}$
3. Compute $err_{(i)} = x_{(i)} - z_{(i-1)}$
4. Compute the estimated variance $\hat{\sigma}_{err}{}^{2}_{(i)} = \vartheta . err_{(i)}{}^{2} + (1 - \vartheta) . \hat{\sigma}_{err}{}^{2}_{(i-1)}$
5. Compute $UCL_{(i)}$ and $LCL_{(i)}$:
6. $UCL_{(i)} = z_{(i-1)} + L . \hat{\sigma}_{err\,(i-1)}$
7. $LCL_{(i)} = z_{(i-1)} - L . \hat{\sigma}_{err\,(i-1)}$
8. *IF* $(LCL_{(i)} < x_{(i)} < UCL_{(i)})$
9. *THEN* (Continue processing with new sample)
10. *ELSE* (Covariate Shift detected and Initiate an appropriate corrective action)

Datasets

Synthetic Data

Dataset 1-Jumping Mean (D1): $x(t) = 0.6\,x(t-1) - 0.5\,x(t-2) + \varepsilon_t$
where ε_t is a noise with mean μ and standard deviation 1.5. The initial values are set as $x(1) = x(2) = 0$.
A change point is inserted at every 100 time steps by setting the noise mean μ at time t as

$$\mu_N = \begin{cases} 0 & N = 1 \\ \mu_{N-1} + \dfrac{N}{16} & N = 2 \ldots .49 \end{cases}$$

Dataset 2-Scaling Variance (D2): The change point is inserted at every 100 time steps by setting the noise standard deviation σ at time t as

$$\sigma = \begin{cases} 1 & N = 1,3, \ldots \ldots, 49 \\ \ln(\mathrm{e} + \dfrac{N}{4}) & N = 2,4, \ldots 48 \end{cases}$$

Dataset 3-Positive-Auto-correlated (D3): The dataset is consisting of 2000 data-points, the non stationarity occurs in the middle of the data stream, shifting from $\mathcal{N}(x; 1,1)$ to $\mathcal{N}(x; 3,1)$, where $\mathcal{N}(x; \mu, \sigma)$ denotes the normal distribution with mean and standard deviation respectively.

Datasets

- ***Real-world Dataset***
  - The real-world data used here are from BCI competition-III dataset (IV-b). This dataset, contains 2 classes, 118 EEG channels (0.05-200Hz), 1000Hz sampling rate which is down-sampled to 100Hz, 210 training trials, and 420 test trials.

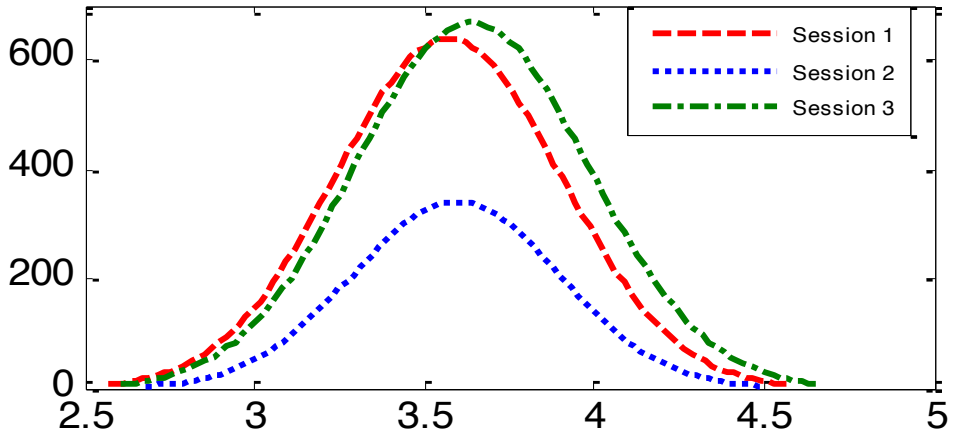
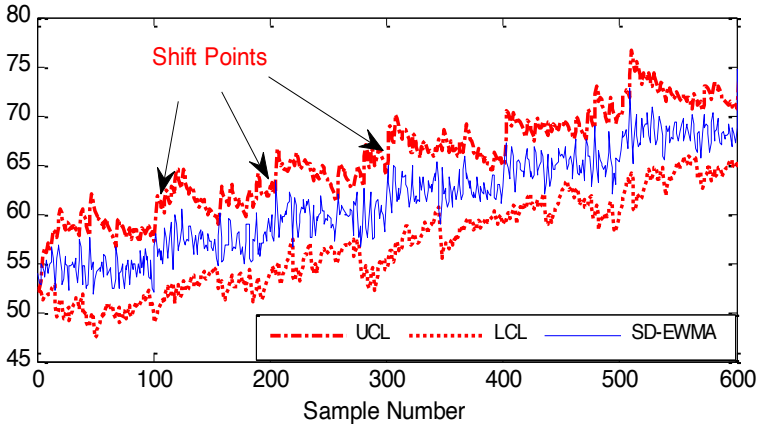


Figure **:** Histogram plot of 3 different sessions' data taken from the training dataset. It is clear from the plot that, in each session the distribution is changed by shifting the mean from session-to-session transfer.



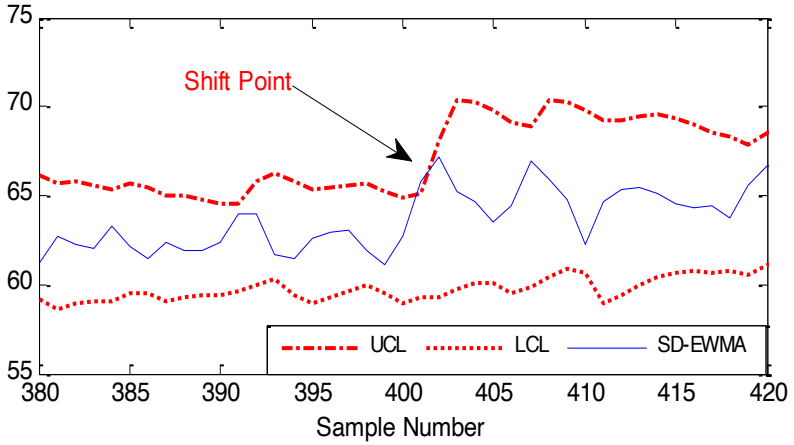


Figure: Shift detection based on SD-EWMA: Dataset 1 (jumping mean): (a) the shift point is detected at every 100th point. (b) Zoomed view of figure a: shift is detected at 401st sample by crossing the upper control limit.

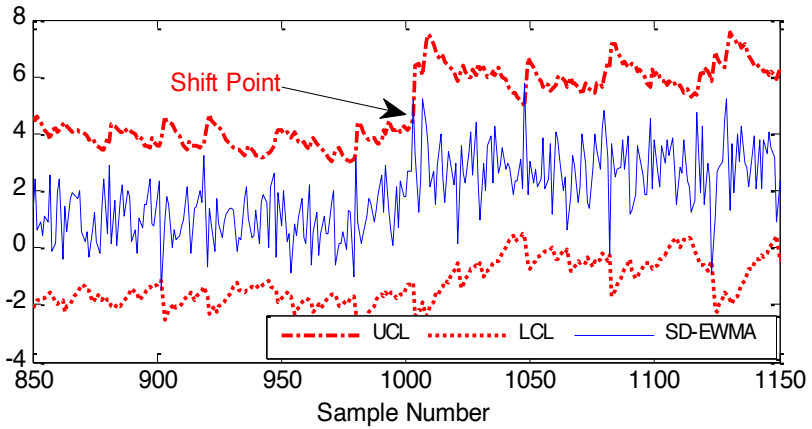


Figure: Shift detection based on SD-EWMA: Dataset 3 (positive auto-correlated): detects the shift after producing 1000 observations.

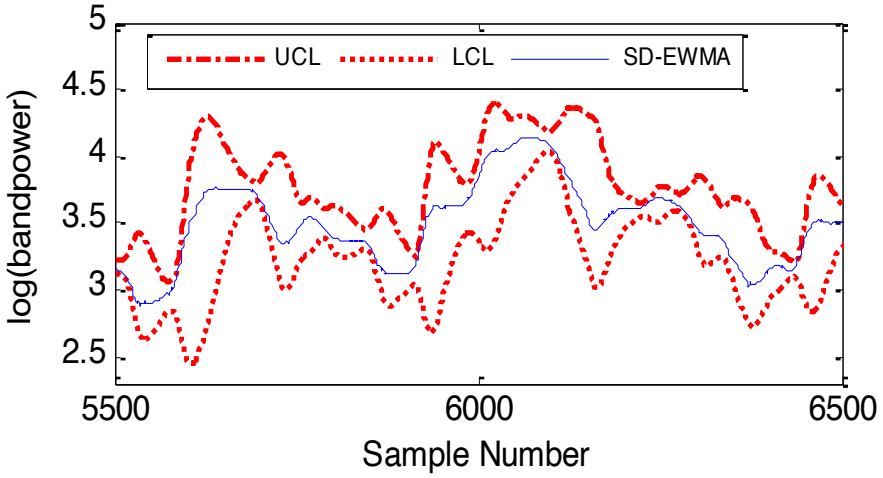
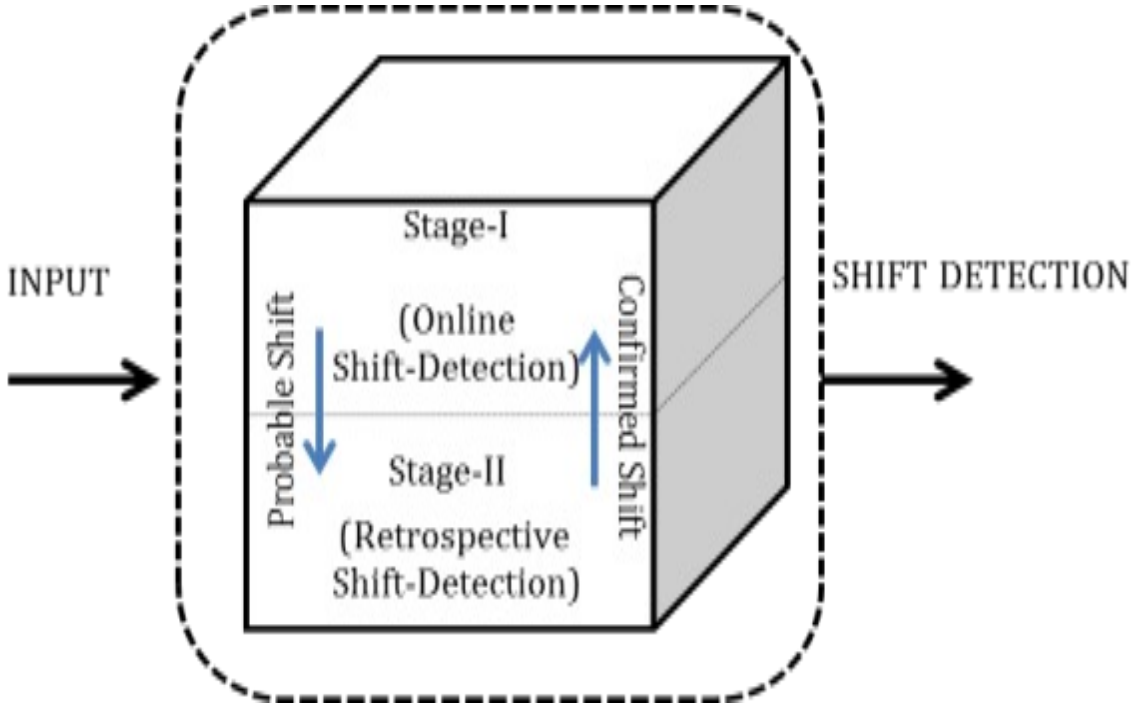


Figure: A window of 2000 samples obtained from real-world dataset.

Two-stage Dataset Shift-detection Based on an Exponentially Weighted Moving Average (TSSD-EWMA)



TSSD-EWMA: Stage-I

Input: Submit the training dataset to the training phase and compute the parameters for testing.
Receive new data in the testing phase sample-by-sample and perform the check as follows.
IF (Shift detected)
THEN (Report the point of shift and move to stage-II for validation)
ELSE (Continue and integrate the upcoming information).

Output: Shift-detection points.

Stage-I

Training Phase

1. Assign training data to $x_{(i)}$ for i=1:n, where n is the number of observations in training data
2. Calculate the mean of $x_{(i)}$ and set as $z_{(0)}$.
3. Compute the z-statistics for each observation $x_{(i)}$ in training data for a range of λ values.

$$z_{(i)} = \lambda . x_{(i)} + (1 - \lambda) . z_{(i-1)}$$

1. Estimate λ by minimizing over the training dataset the square of 1-step-ahead prediction error: $err_{(i)} = x_{(i)} - z_{(i-1)}$.
2. Finally estimate the variance of error for the testing phase.

TSSD-EWMA: Stage-I

Stage-I

Testing Phase

1. For each data point $x_{(i)}$ in the operation/testing phase
2. Compute $z_{(i)} = \lambda . x_{(i)} + (1 - \lambda) . z_{(i-1)}$
3. Compute $err_{(i)} = x_{(i)} - z_{(i-1)}$
4. Estimate the variance $\hat{\sigma}_{err}^{2}_{(i)} = \vartheta . err_{(i)}^{2} + (1 - \vartheta) . \hat{\sigma}_{err}^{2}_{(i-1)}$
5. Compute $UCL_{(i)}$ and $LCL_{(i)}$:
6. $UCL_{(i)} = z_{(i-1)} + L . \hat{\sigma}_{err\,(i-1)}$
7. $LCL_{(i)} = z_{(i-1)} - L . \hat{\sigma}_{err\,(i-1)}$
8. *IF* $(LCL_{(i)} < x_{(i)} < UCL_{(i)})$
   THEN (Continue processing)
   ELSE (Go to Stage-II)

TSSD-EWMA: Stage-II

Stage-II

1. For each $x_{(i)}$
2. Wait for m observations after the time i, organise the sequential observations around time i into two partitions, one containing $x_{((i-(m-1)):i)}$, another $x_{((i+1):(i+m))}$.
3. Execute the hypothesis test on the partitioned data
4. IF$(H=1)$

THEN (test rejects the null hypothesis): Alarm is raised

ELSE(The detection received by stage-I is a false and discarded)

Table: TSSD-EWMA shift-detection

	Total shifts	Lambda (λ)	SD-EWMA # TP	SD-EWMA # FP	SD-EWMA # FN	SD-EWMA ACC	TSSD-EWMA # TP	TSSD-EWMA # FP	TSSD-EWMA # FN	TSSD-EWMA AD	TSSD-EWMA ACC
D1	9	0.20	8	1	1	99.8	8	0	1	10	99.9
		0.30	8	3	1	99.6	8	0	1	10	99.9
		0.40	All	4	0	99.6	All	0	0	10	100
D2	5	0.60	All	11	1	99.8	NA	NA	NA	NA	NA
		0.70	All	6	0	99.4	NA	NA	NA	NA	NA
		0.80	4	8	1	99.1	NA	NA	NA	NA	NA
D3	1	0.40	All	13	0	99.1	All	0	0	10	100
		0.50	All	12	0	99.2	All	0	0	10	100
		0.60	All	15	0	99.0	All	0	0	10	100

Table: TSSD-EWMA shift-detection in BCI data

Lambda (λ)	Number of Trials	# Shift-points SD-EWMA Session 2	# Shift-points SD-EWMA Session 3	# Shift-points TSSD-EWMA Session 2	# Shift-points TSSD-EWMA Session 3
0.01	1-20	*0*	0	0	0
	21-45	3	1	2	0
	46-70	4	2	3	1
0.05	1-20	*10*	4	6	3
	21-45	9	6	6	4
	46-70	7	5	6	4
0.10	1-20	16	9	8	6
	21-45	22	21	15	15
	46-70	25	17	17	13

Adaptive Learning with Covariate Shift–Detection (ALCSD)

- Is a single classifier based non-stationary learning (NSL) algorithm.
- Is an active learning, where the learning model updates on each covariate shift-detection.

The key elements of the proposed solution are:

- $\boldsymbol{SDT_X}$: the SDT (shift detection test) analyses the raw observations to monitor the stationarity of x_i, disregarding their supervised labels.
- $\boldsymbol{K}$: The base classifier used to classify the input samples.
- $\boldsymbol{KB_{Updated}}$: Updated knowledge base (KB) using the data with covariate shift.

(ALCSD) Cont.....

Algorithm: ALCSD

Configure the classifier K based on the initial knowledge base KB_0;
Configure the SDT_X using the initial knowledge base KB_0 ;
FOR $i = 1$ to the length of testing data
  Receive new data x_i;
  IF (SDT_X detects a non-stationarity at time i), **THEN**
    Update the knowledge base (KB) for classifier K to $KB_{Updated}$;
    Retrain the classifier on $KB_{Updated}$ as suggested in the Table I
  END
  Classify the input x_i by classifier K and get the predicted label $\hat{y}_i$;
END

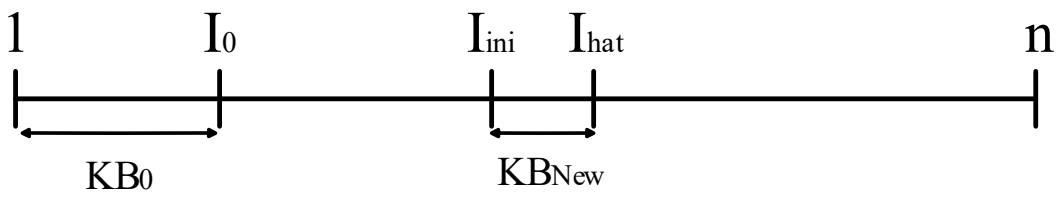


Table I. Proposed Cases

No.	$KB_{Updated}$: Method to update KB and retrain classifier
A	*Learning without CSD*
B	*Adaptive learning with CSD*
C	*Adaptive learning on KB_{New} with CSD*
D	*Adaptive learning on combined KB with CSD*
E	*Transductive learning with CSD*

Experiments and Results

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

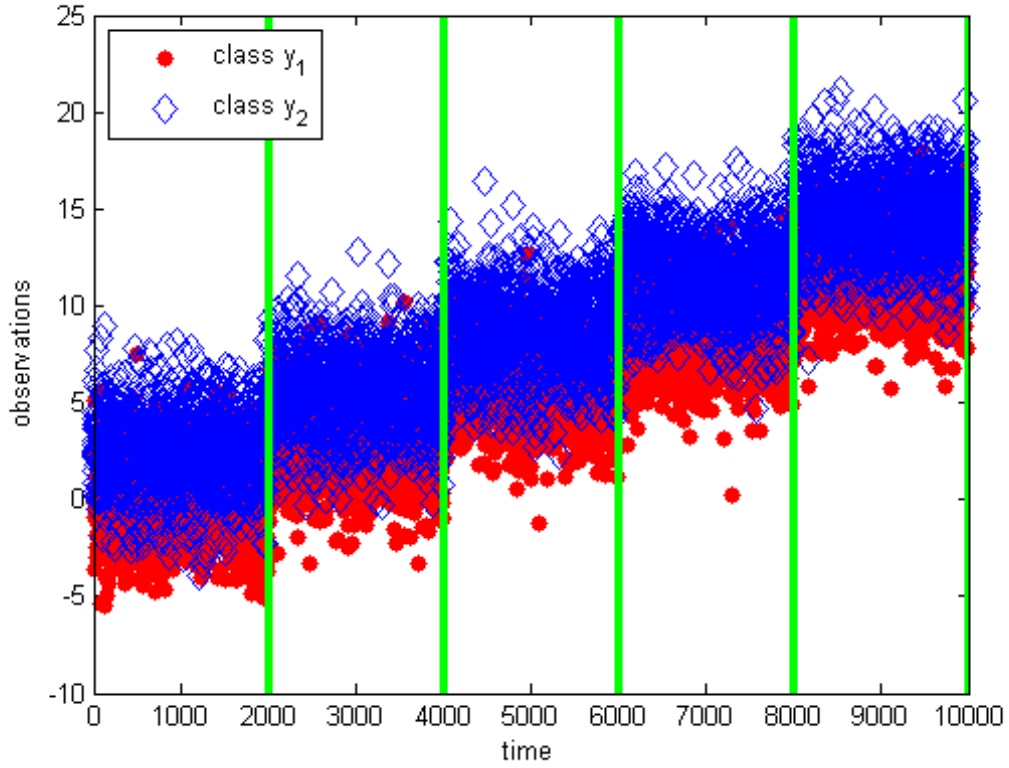
Datasets*

- Dataset 1-One class covariate shift
- Dataset 2-Classes swap concept shift
- Dataset 3-Abrupt covariate shift affecting both classes
- Dataset 4- Transient covariate shift
- Dataset 5-Altering concepts shift and classes Swap
- Dataset 6- Stairs sequence of covariate shifts

Evaluation

- The classification error rate

- Classifiers Used:
  - k-nearest neighbor (k-NN)
  - Support Vector Machine (SVM)
  - Naïve Bayes (NB)



* C. Alippi, G. Boracchi, and M. Roveri, "Just-In-Time Classifiers for Recurrent Concepts," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 24, no. 4, pp. 620–634, Apr. 2013.

Table II: Classification Error rate for different datasets using k-NN Classifier

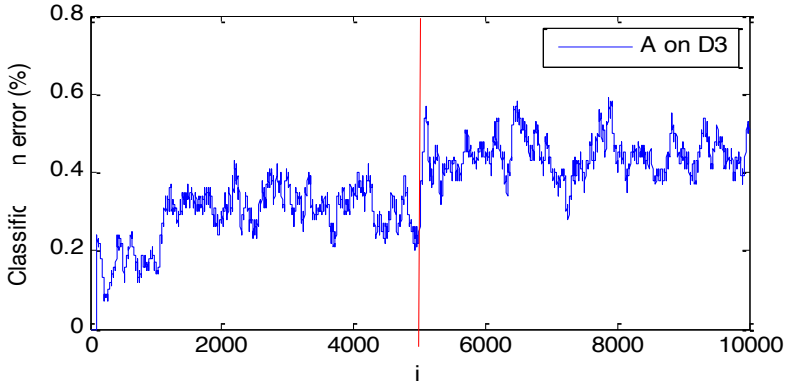
	A	B		C		D		E	
		Pred	Sup	Pred	Sup	Pred	Sup	Pred	Sup
D1	0.2414	0.2417	0.2385	**0.2316**	**0.1989**	0.2404	0.2345	0.2476	0.2373
D2	0.4806	0.4807	0.4807	0.4807	0.4807	0.4807	0.4807	0.4807	0.4807
D3	0.3301	0.3312	0.3293	**0.3243**	**0.2987**	0.3312	0.3367	0.3359	0.3343
D4	0.3190	0.3176	0.3173	0.3151	0.3322	0.3364	0.3309	0.3104	0.3295
D5	0.4492	0.4496	0.4496	0.4496	0.4496	0.4496	0.4496	0.4496	0.4496
D6	0.4290	0.4290	0.3789	**0.4139**	**0.3154**	0.4209	0.3745	0.4227	0.3726

Table III: Classification Error rate for different datasets using SVM Classifier

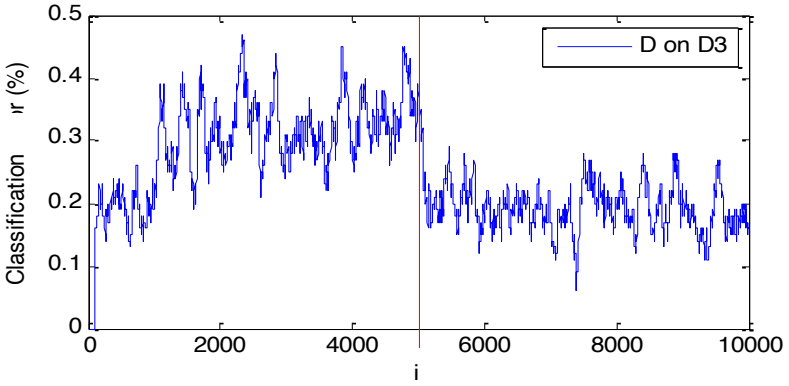
	A	B		C		D		E	
		Pred	Sup	Pred	Sup	Pred	Sup	Pred	Sup
D1	0.2452	0.2468	0.2584	**0.2352**	**0.2328**	0.2419	0.238	0.234	0.232
D2	0.4946	0.4946	0.4946	0.4946	0.4946	0.4946	0.4946	0.4946	0.4946
D3	0.3114	0.3151	0.3201	0.3218	0.3125	0.3133	0.3164	0.3133	0.3205
D4	0.3002	0.2891	0.3044	0.2841	0.2821	0.2889	0.2764	0.2861	0.3043
D5	0.4618	0.4621	0.4621	0.4621	0.4621	0.4621	0.4621	0.4621	0.4621
D6	0.4359	0.4158	0.4091	**0.2887**	**0.2798**	0.4199	0.3338	0.411	0.3321

Table IV: Classification Error rate for different datasets using NB Classifier

	A	B		C		D		E	
		Pred	Sup	Pred	Sup	Pred	Sup	Pred	Sup
D1	0.2157	0.2107	0.2068	**0.2002**	**0.1782**	0.1956	0.1787	0.1962	0.1789
D2	0.4946	0.4946	0.4946	0.4946	0.4946	0.4946	0.4946	0.4946	0.4946
D3	0.3103	0.3099	0.3005	**0.3021**	**0.2687**	0.3161	0.2698	0.3144	0.2697
D4	0.3004	0.3005	0.296	0.3086	0.2835	0.3009	0.2769	0.3017	0.277
D5	0.4539	0.4543	0.4543	0.4543	0.4543	0.4543	0.4543	0.4543	0.4543
D6	0.4356	0.4348	0.4318	**0.4328**	**0.28**	0.4378	0.3475	0.4363	0.3473



(a)



(b)

Figure 5. The classification error as function of time for the propose method vs. traditional method. The average classification error is computed on a window containing the 100 supervised samples.
(a) Classification error with method A on dataset D3.
(b) Classification error with method D on dataset D3. The red line shows the point where the shift has occurred.

Conclusion

- We presented a two-stage structure for covariate shift-detection (**TSSD-EWMA**)
  - The first stage works in an online mode, and it uses an exponentially weighted moving average (SD-EWMA) model-based control chart to detect the covariate shift-point.
  - The second stage validates the shift-detected by first stage using statistical hypothesis test

- The experimental results shows
  - Advantage: Two stage reduces false-positive
  - Disadvantage: Time delay due to data accumulation for validation

- **TSSD-EWMA** is used to develop an adaptive learning framework for non-stationary learning
  - Adaptive Learning with Covariate Shift–Detection (**ALCSD**)

- **ALCSD** is computationally efficient because it does not require an additional memory to store all the observations. Only the initial knowledge-base (KB_0) and a small window of KB_{New} are to be kept.

- R-package

  otsad: Online Time Series Anomaly Detectors

  cran.r-project.org/web/packages/otsad/index.html

Acknowledgement

This research was carried out in collaboration with

Dr Haider Raza

Prof Girijesh Prasad