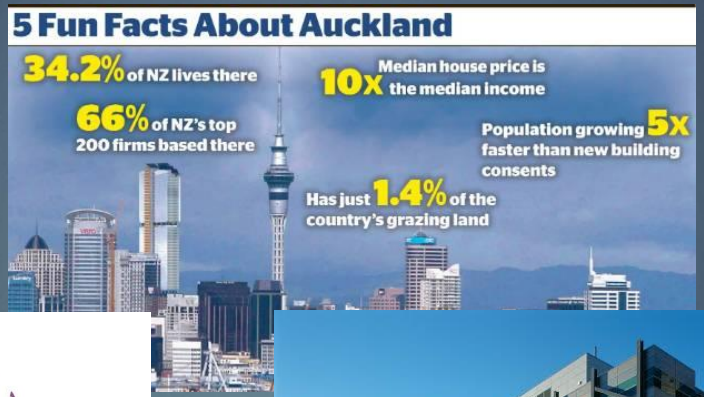
Using Volatility in Concept Drift Detection and Capturing Recurrent Concept Drift in Data Streams

YUN SING KOH

ykoh@cs.auckland.ac.nz

https://www.cs.auckland.ac.nz/~yunsing/

Where is Auckland?



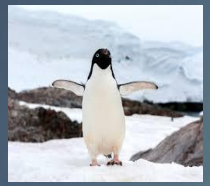
Data Mining Task

- Prediction Tasks
  - Use some variables to predict unknown or future values of other variables
- Description Tasks
  - Find human-interpretable patterns that describe the data.

Common data mining tasks includes:

- Classification [Predictive]
- Clustering [Descriptive]
- Association Rule Discovery [Descriptive]
- Sequential Pattern Discovery [Descriptive]
- Regression [Predictive]
- Deviation Detection [Predictive]

Predictive – Classification

x							
f(x)	zebra	penguin	zebra	penguin	zebra	zebra	?

Data Streams

- **Data Stream Mining** is the process of extracting knowledge structures from continuous, rapid data records. A data stream is an ordered sequence of instances that in many applications of data stream mining can be read only once or a small number of times using limited computing and storage capabilities.

Properties

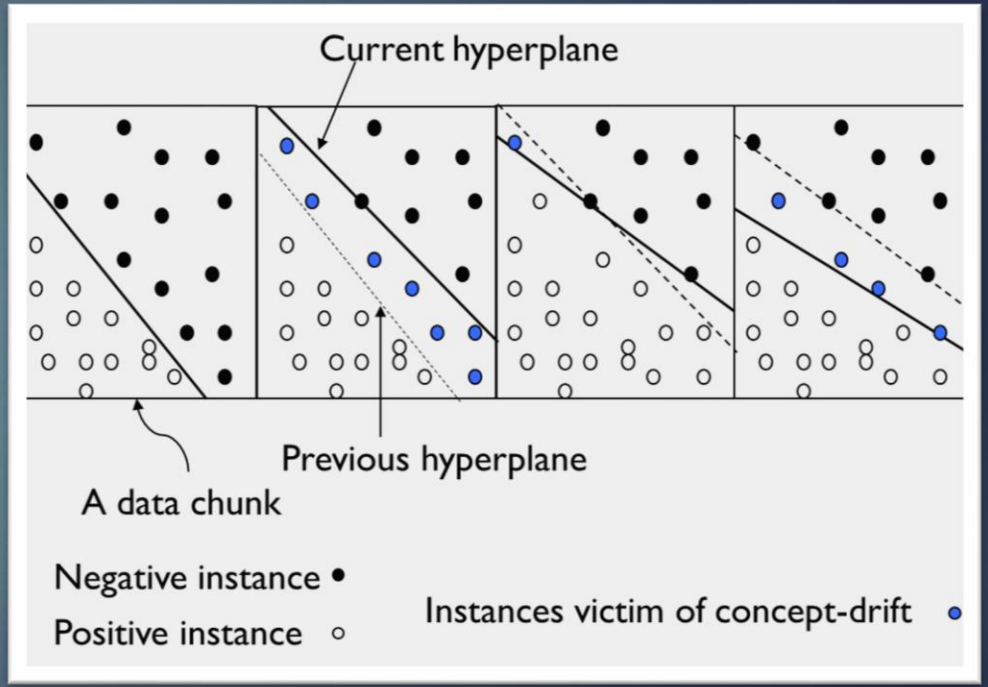
1. At high speed
2. Infinite
3. Can't store them all
4. Can't go back; or too slow
5. Evolving, non-stationary reality

What this means in an algorithmic sense?

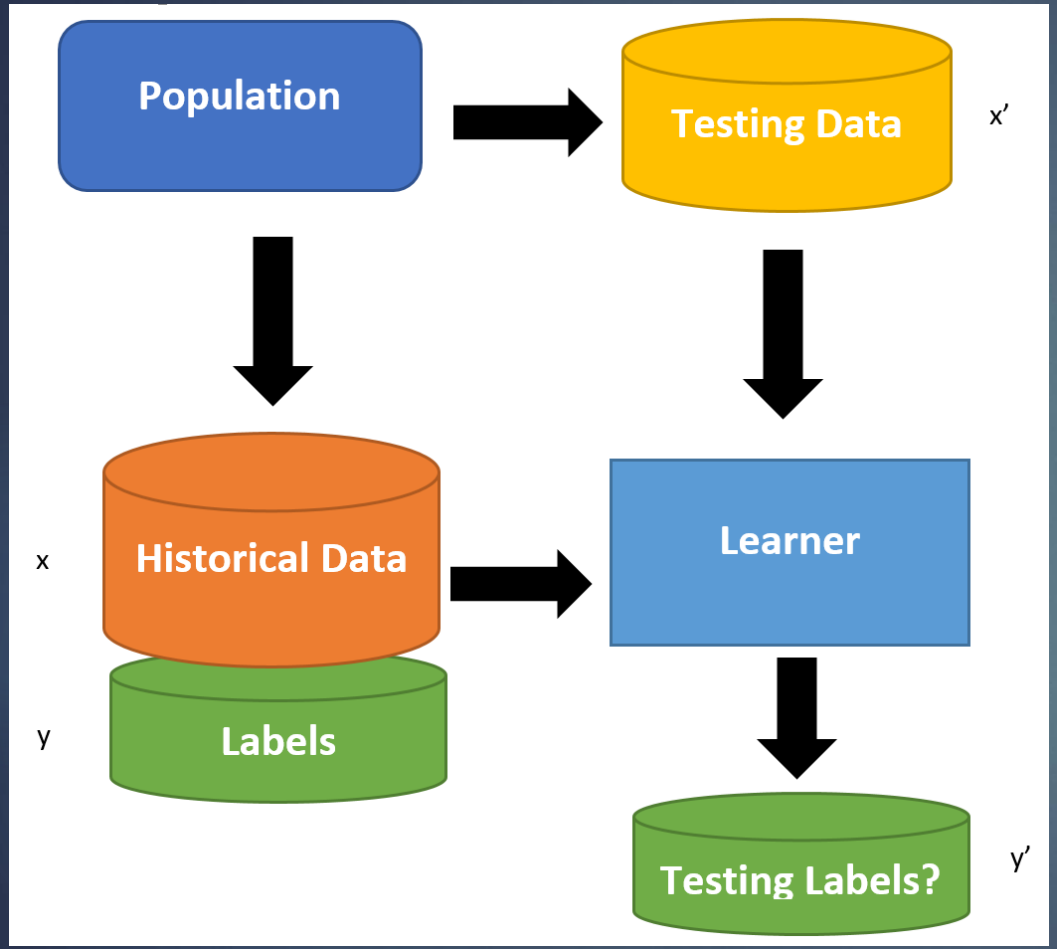
1. One pass
2. Low time per item - read, process, discard
3. Sublinear memory - only summaries or sketches
4. Anytime, real-time answers
5. The stream evolves over time

Volume, Velocity, Variety & **Variability**

- data comes from complex environment, and it evolves over time.
- concept drift = underlying distribution of data is changing



$$\exists X : p_{t_0}(X, y) \neq p_{t_1}(X, y),$$



Training:
Learning a mapping function

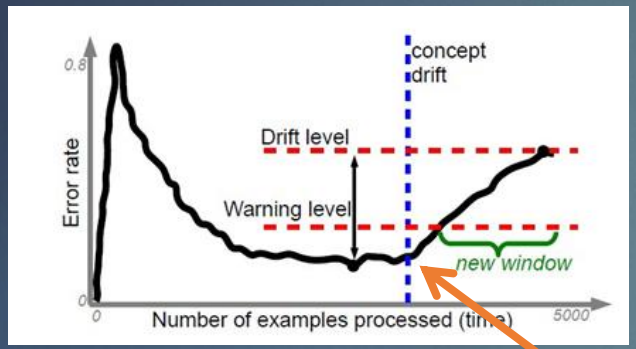
$$y = f(x)$$

Application:
Applying *f* to unseen data

$$y' = f(x')$$

Supervised Learning

Concept Drift & Error rates



- When there is a change in the class-distribution of the examples:
  - The actual model does not correspond any more to the actual distribution.
  - The error-rate increases
- Basic Idea:
  - Learning is a process.
  - Monitor the quality of the learning process:
    - Monitor the evolution of the error rate.

Adaptation Methods

- The Adaptation model characterizes the changes in the decision model do adapt to the most recent examples.

- Blind Methods:
  - Methods that adapt the learner at regular intervals without considering whether changes have really occurred.

- Informed Methods:
  - Methods that only change the decision model after a change was detected. They are used in conjunction with a detection model.

Background - Concept Drift

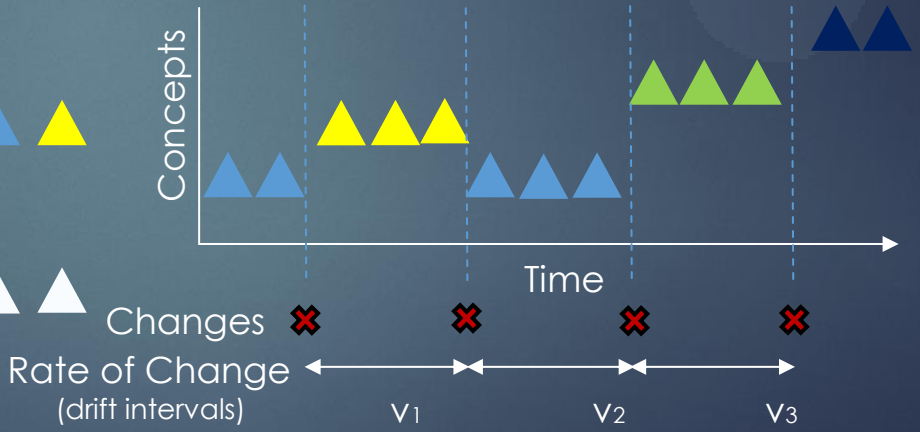
Types of drift

1. Abrupt
2. Gradual
3. Incremental

Drift Volatility

- Rate of concept change

Example



SEED Detector – Change Detector

David Tse Jung Huang, Yun Sing Koh, Gillian Dobbie, Russel Pears: Detecting Volatility Shift in Data Streams. ICDM 2014

- As each instance of the data (predictive error rates) arrives it is stored in a block B_i each block can store up to x number of instances.
- To check for drift, the window W is split into two sub-windows W_L and W_R and each of the boundaries between the blocks is considered as a potential drift.

$$W = B_1 \mid B_2\ B_3\ B_4\ B_5$$
$$W = B_1\ B_2 \mid B_3\ B_4\ B_5$$
$$W = B_1\ B_2\ B_3 \mid B_4\ B_5$$
$$W = B_1\ B_2\ B_3\ B_4 \mid B_5$$

$$|\mu W_L - \mu W_R|$$

- Using every boundary as potential drift point is excessive. SEED performs block compressions to merge consecutive blocks that are homogeneous in nature.

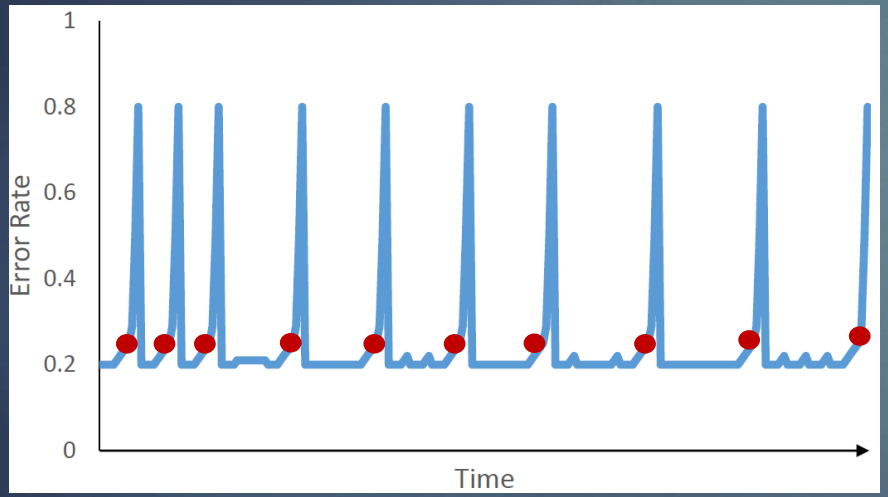
Volatility Shift in Data Streams

David Tse Jung Huang, Yun Sing Koh, Gillian Dobbie, Russel Pears: Detecting Volatility Shift in Data Streams. ICDM 2014

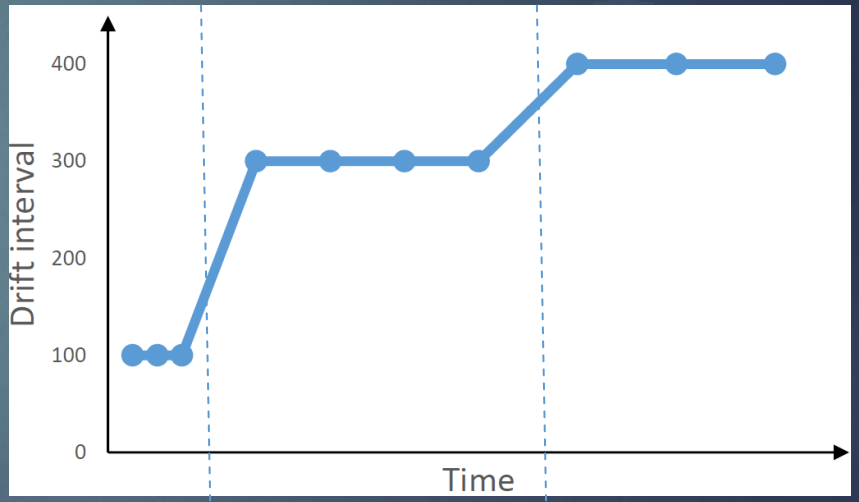
- It is useful to understand characteristics of a stream, such as volatility.
- Example: Machine performance and maintenance
  - Drift: Deviations in machine performance.
  - Volatility: Monitoring the deviations.

Example of Drift Volatility

- Error rate stream showing drift points

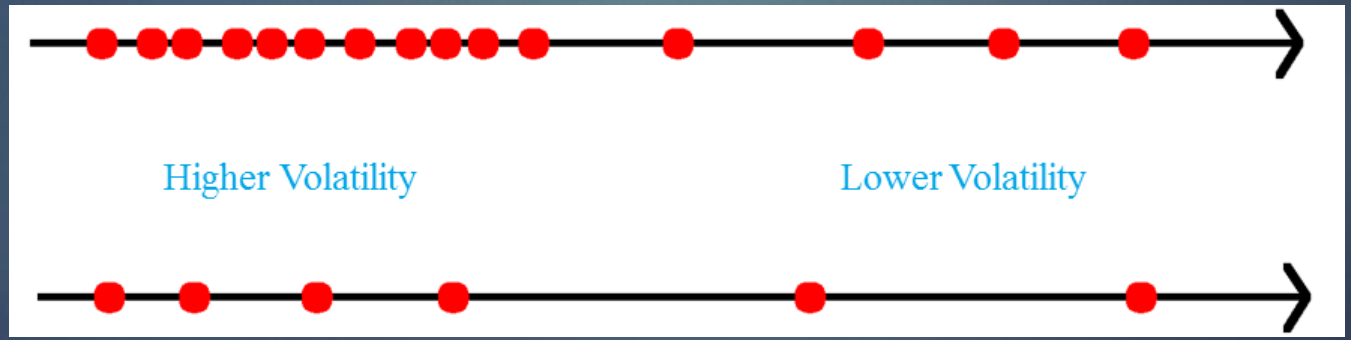


- Drift volatility (rate of change)



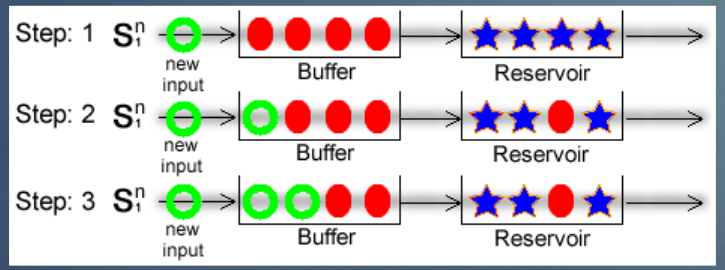
Volatility Shift in Data Streams

- A stream has a high volatility if drifts are detected frequently and has a low volatility if drifts are detected infrequently.
- Streams can have similar characteristics but be characterized as stable and non-volatile in one field of application and extremely volatile in another.



Volatility Detector Example

- There are two main components in our volatility detector: a buffer and a reservoir.
- The buffer is a sliding window that keeps the most recent samples of drift intervals acquired from a drift detection technique.
- The reservoir is a pool that stores previous samples which ideally represent the overall state of the stream.



$$Relative\ Volatility = \frac{\sigma_{BUFFER}}{\sigma_{RESERVOIR}}$$

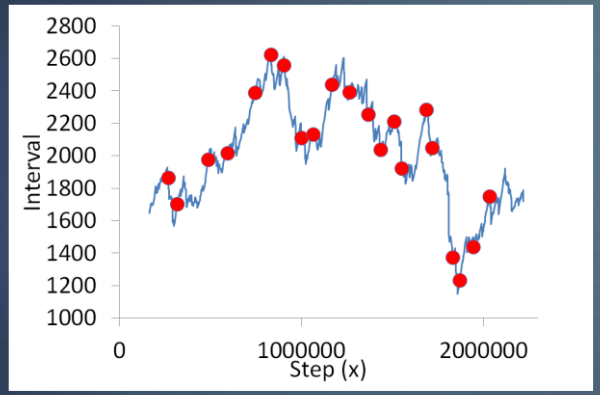
Shift in Relative Variance:

Given a user defined confidence threshold $\beta \in [0,1]$, a shift in relative variance occurs when

$$Relative\ Variance > 1.0 + \beta$$
$$Relative\ Variance < 1.0 - \beta$$

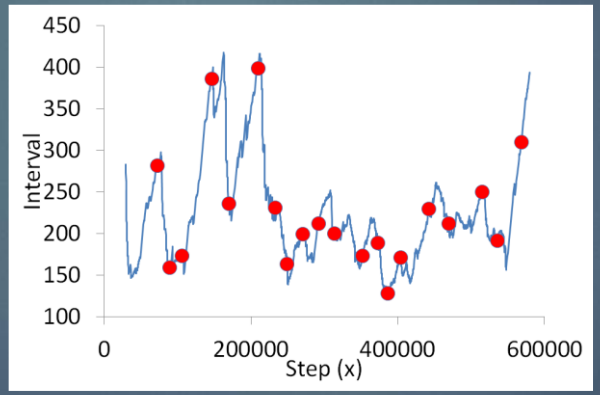
Real World Results

Each stream was evaluated using a Hoeffding Tree to produce the binary stream that represents the classification errors then passed to our drift detector.



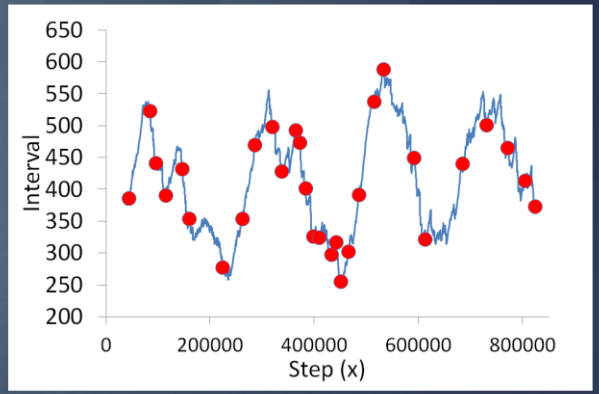
Sensor Stream

- 1,150 change points found
- 21 volatility shifts
- intervals between 1500 to 2500



Forest Covertype

- 2,611 change points found
- 20 volatility shifts
- intervals between 100 to 450



Poker Hand

- 2,059 change points were found
- 30 volatility shifts
- intervals between 150 to 650

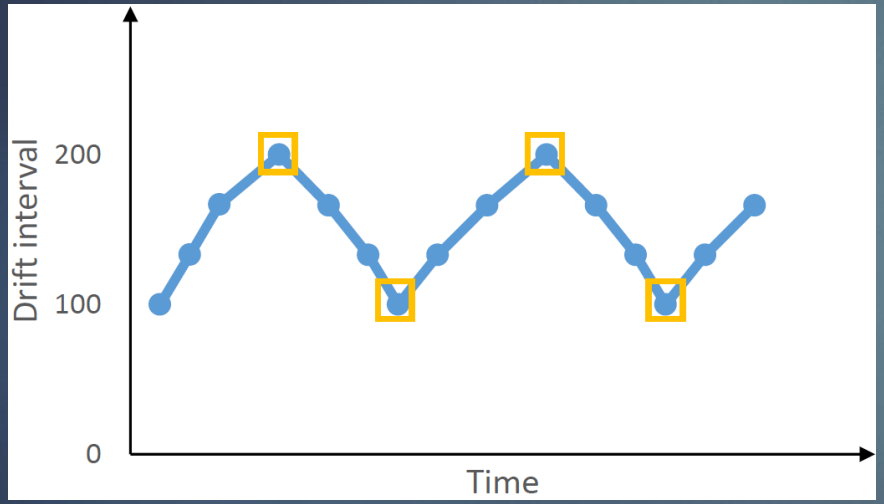
Proactive Drift Detection System

Kylie Chen, Yun Sing Koh, Patricia Riddle: Proactive drift detection: Predicting concept drifts in data streams using probabilistic networks. IJCNN 2016: 780-787

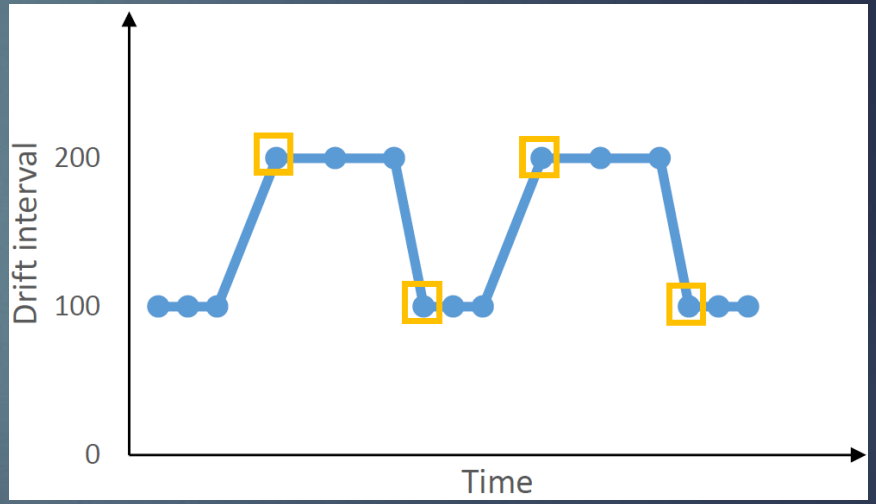
- Modelling Drift Volatility Trends
- Goals:
  - Predict location of next drift
    - Drift Prediction Method using Probabilistic Networks
  - Use predictions to develop proactive drift detection methods
    - Adaptation of Drift Detection Method SEED
    - Adaptation of data structure using compression

Modelling Drift Volatility Trends

- Progressive volatility change



- Rapid volatility change



Example of Drift Prediction Method

Example of drift intervals

100 100 | 100 **300** | 300 300 | 300 **400** | 400 400

1. Identify volatility change points (Volatility Detector)
2. Outlier removal to construct pattern from drift interval windows

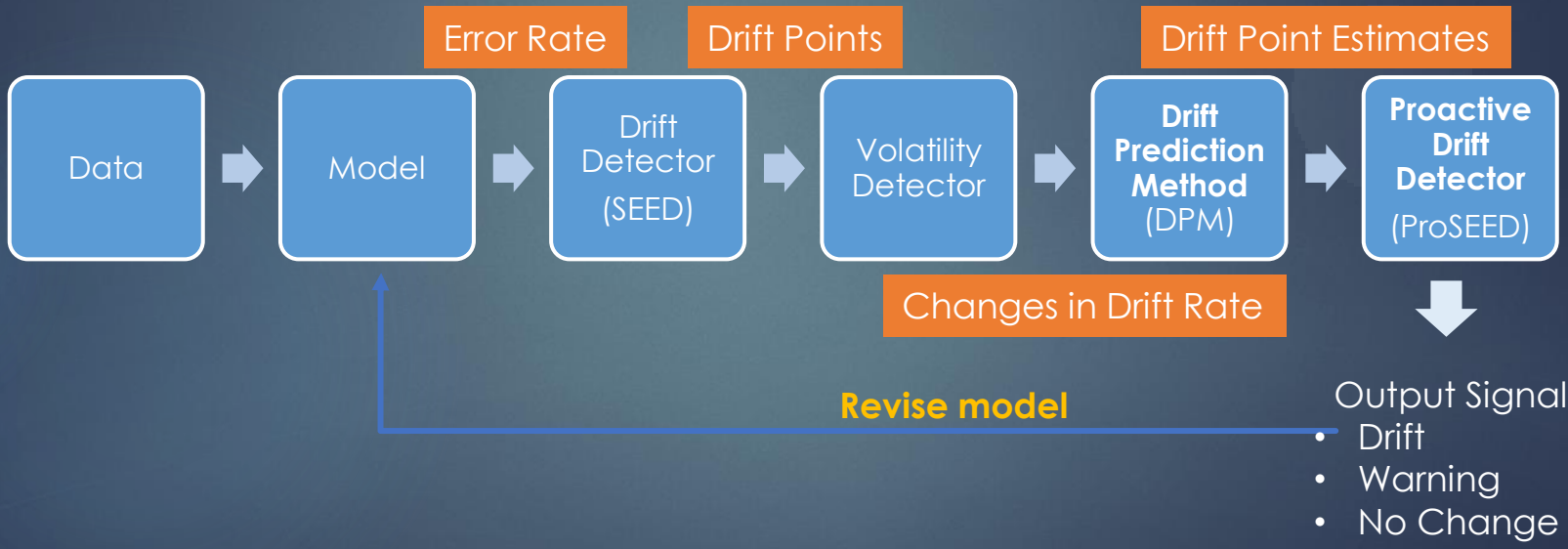
p_2 300 300 300 ? Pattern Reservoir

p_1 100 100 100

p_2 300 300 300

3. Match patterns to stored patterns
4. Update probabilistic network

Proactive Drift Detection System



Adapting the data structure of SEED

Extend the SEED Detector to use predicted drifts from our Drift Prediction Method

Adaptation of data compression of SEED detector

- no compression in blocks where we expect drift

Example of error stream

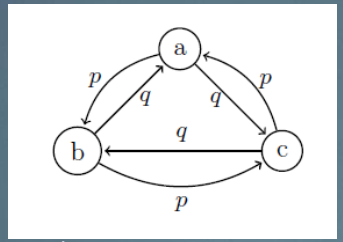
- 00011000100110110111

Expected Predicted drifts at time steps 6 and 18

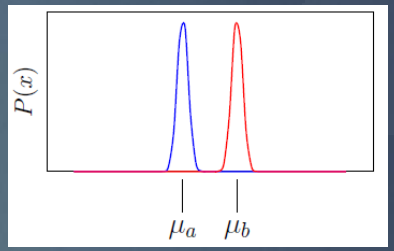
- 0001 | 1000 | 1001 | 1011 | 0111
- c1 c2 c3 c4
- 0001 | 1000 | 10011011 | 0111
- c1 c2 c3

Summary of Datasets

- **Synthetic datasets**
  - Bernoulli
  - SEA Concepts
  - CIRCLES

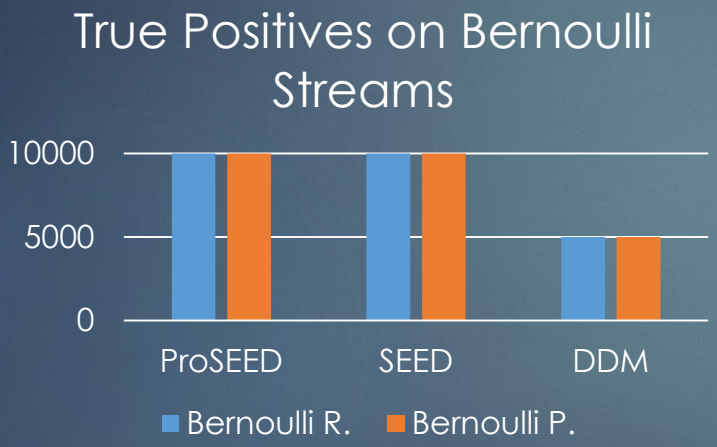


  - Generated with cyclic trends
  - Drift interval distributions generated using Normal Distributions
  - 10,000 drifts per stream
  - 100 trials



- **Real datasets**
  - Forest Covertype
  - Sensor Stream

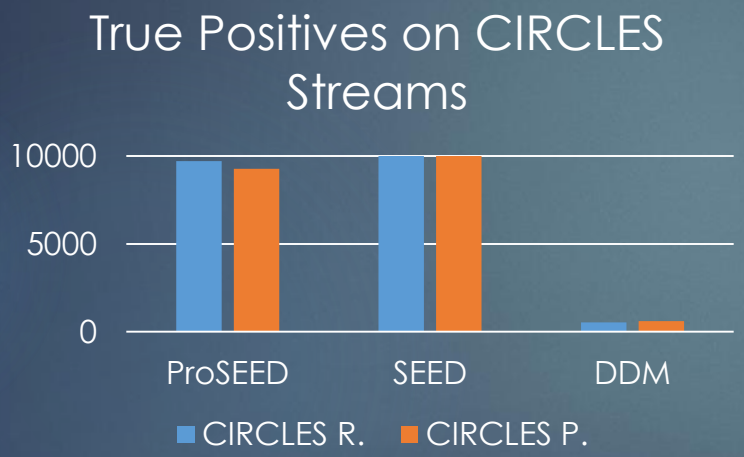
Results - Proactive Drift Detection (Bernoulli)



Average Number of False Positives

Detector	Bernoulli R.	Bernoulli P.
ProSEED	**33.10**	**44.32**
SEED	213.34	210.50
DDM	97.41	100.98

Results - Proactive Drift Detection (CIRCLES)



Average Number of False Positives

Detector	CIRCLES R.	CIRCLES P.
ProSEED	**271.44**	**10.05**
SEED	481.77	531.62
DDM	306.94	380.32

Concept Profiling Framework (CPF)

Robert Anderson, Yun Sing Koh, Gillian Dobbie: CPF: Concept Profiling Framework for Recurring Drifts in Data Streams. Australasian Conference on Artificial Intelligence 2016: 203-214

- Concept Profiling Framework (CPF), a meta-learner that uses a concept drift detector and a collection of classification models to perform effective classification on data streams with recurrent concept drifts, through relating models by similarity of their classifying behaviour.
- Existing state-of-the-art methods for recurrent drift classification often rely on resource-intensive statistical testing or ensembles of classifiers (time and memory overhead that can exclude them from use for particular problems)

Recurring Concept Drifts Models

The Concept Profiling Framework

- A meta-learning framework that can:
  - use observed model behaviour over time to accurately recognise recurring concepts
  - A meta-learning approach that maintains a collection of classifiers and uses a drift detector
  - When drift is detected, either an existing classifier is reused or a new model is added
  - Where classifiers behave similarly, the older will represent the new one
  - We use a fading mechanism to remove models that are not recent nor being reused

Recurring Concept Drift

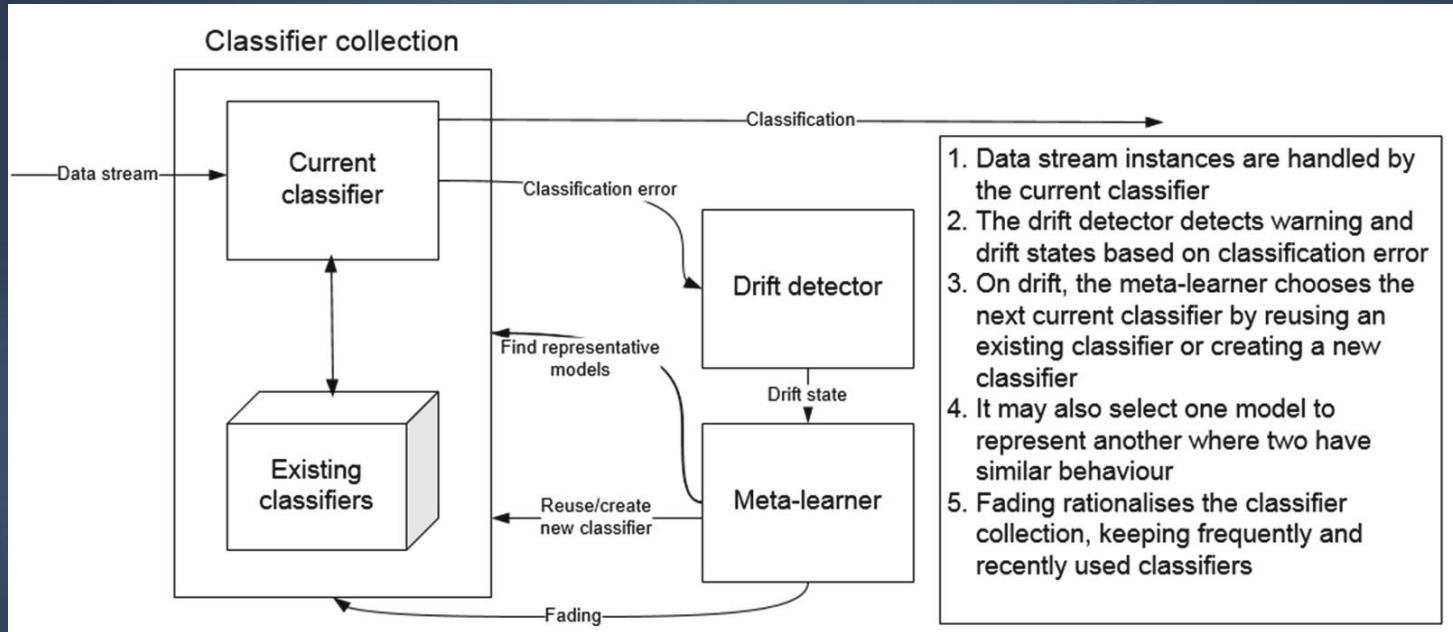


Fig. 1. The concept profiling framework

Model testing and reuse

- At every detected drift point, we test all models on the warning buffer. We always create a new model unless:
  - An existing model gets an accuracy of *m* (CPF's similarity parameter) on the warning buffer OR
  - An existing model gets a similarity of *m* to the newly trained model

Similarity Between Models

	Errors on warning buffer instances (X)										Accuracy on buffer	Sim(X, c_1, c_n)	Sim(X, c_{new}, c_n)
Classifier (c_n)	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}			
Classifier 1 (c_1)	0	1	0	0	0	1	0	0	1	1	6 / 10	10 / 10	4 / 5
Classifier 2 (c_2)	1	1	0	0	1	1	0	1	1	1	3 / 10	7 / 10	2 / 5
Classifier 3 (c_3)	0	0	0	0	0	1	0	0	0	0	9 / 10	7 / 10	5 / 5
New classifier (c_{new}) **T = training instances*	0	T	0	T	0	T	0	T	0	T	5 / 5	4 / 5	5 / 5

Fig. 2. Example of calculating model similarity for classifiers $c_1 ... c_{new}$ on warning buffer X

Representation - building a picture of model similarity

- When models behave similarly on warning buffer instances i.e. have a score ≥ m, we keep the older model to represent the newer model.
- This speeds up the procedure and allows us to identify recurring concepts.
- We track similarity over time between models: eventually models based on the same concepts should look similar and pass the m threshold.

Fading

- Used to increase efficiency of our technique by keeping the classifier collection small.

1. When a model is created or reused, it gets **f** points.
2. Every drift where it is not reused, it loses a point.
3. When it has zero points, it is deleted.
4. If a model is chosen to represent another, their fade points are combined.

Fading mechanism

- Our technique very commonly worked significantly faster and with less memory, while maintaining accuracy using our proposed fading mechanism.
- The model collection was restrained through use of our fade mechanism.

Dataset	Approach	Mean accuracy	Mean memory	Mean time	Mean models	Max models
Agrawal	CPF	78.18%	**9.91E+05**	**2.60E+04**	16.94	21.83
	CPF - no fade	**80.00%**	2.70E+08	1.54E+06	1183.04	2153.47
CIRCLES	CPF	96.39%	**9.42E+05**	**5.15E+03**	9.75	18.93
	CPF - no fade	**97.18%**	2.05E+06	7.43E+03	44.52	55.50
Hyperplane	CPF	**93.61%**	**5.15E+05**	**2.13E+04**	17.04	24.37
	CPF - no fade	92.90%	1.25E+08	5.74E+05	813.68	1500.77
LED	CPF	69.10%	**5.90E+05**	**6.92E+04**	16.80	18.40
	CPF - no fade	68.95%	6.19E+07	1.15E+06	491.48	971.43
RandomRBF	CPF	80.32%	**1.32E+06**	**2.99E+04**	16.21	21.60
	CPF - no fade	81.13%	1.64E+08	9.43E+05	797.00	1477.77
SEA	CPF	84.62%	**1.76E+06**	**9.02E+03**	13.20	20.10
	CPF - no fade	84.46%	5.06E+06	1.31E+04	39.87	60.87
STAGGER	CPF	99.33%	4.30E+05	3.05E+03	4.05	4.83
	CPF - no fade	99.33%	4.32E+05	2.98E+03	4.27	4.87
Waveform	CPF	78.15%	**1.15E+06**	**1.00E+05**	16.47	26.47
	CPF - no fade	78.36%	9.92E+07	1.17E+06	303.87	565.87

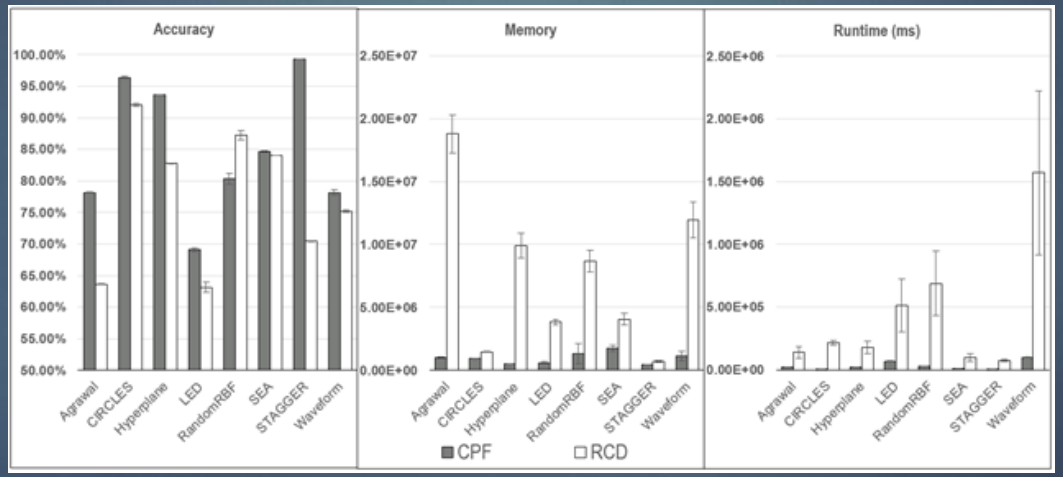
Similarity margin to use

- Our technique was rated on how well it did against all datasets with different minimum similarity margin to reuse a model or represent one with another
- We wanted a setting that did most consistently across all datasets i.e. had fewest bad rankings

Similarity margin *(m)*	Rank on synthetic datasets (lower is better)									Rank on real datasets					
	Agrawal	CIRCLES	Hyperplane	LED	Random RBF	SEA	STAGGER	Waveform	Bad rankings	Electricity	Poker	Intrusion	Airlines	Social Media	Bad rankings
0.85	2	5	5	5	1	5	5	5	6	5	5	4	5	5	5
0.90	1	4	4	4	2	4	4	3	5	3	4	1	1	4	2
0.95	3	1	1	3	3	3	3	4	1	4	1	3	3	3	1
0.975	4	2	2	1.5	4	2	2	1	2	2	2	2	3	2	0
0.99	5	3	3	1.5	5	1	1	2	2	1	3	5	3	1	1

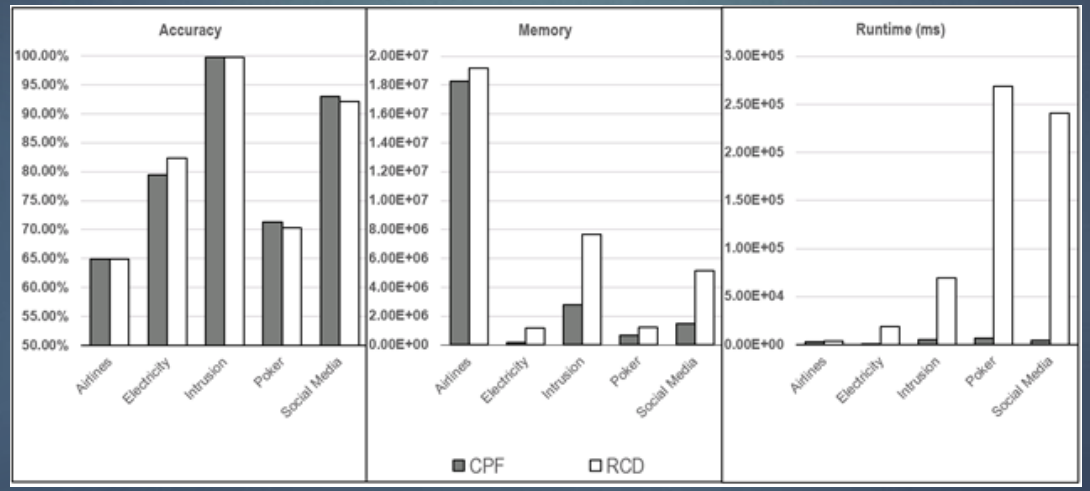
Synthetic datasets

- Our technique generally achieved better accuracy while taking less time and memory than RCD



Real-world datasets

- Our technique generally maintained similar accuracy while taking less time and memory than RCD



Thank you and Questions?