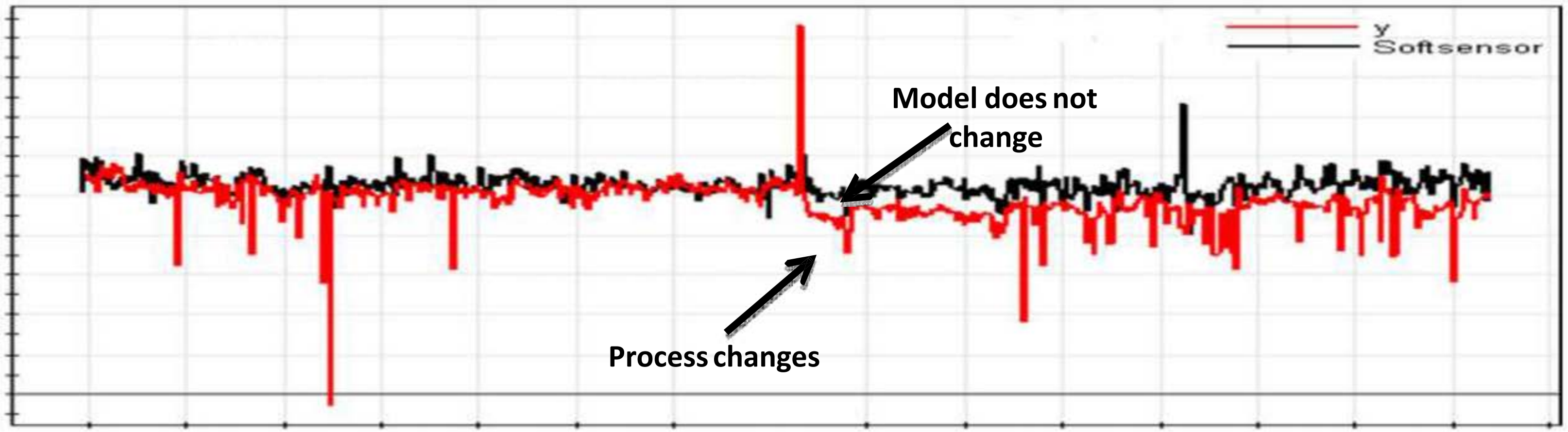
Concept Drift

the supplier of raw material changes

a sensor breaks/ “wears off”/ is replaced

new operating crew



source: Evonik Industries

new regulations to save electricity

new production procedures

Concept drift

- Concept drift between time point t0 and time point t1 can be defined as:

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

- The goals:
- to introduce the problem of concept drift in supervised learning
- to overview and categorize the main principles how concept drift can be (and is) handled

Types of changes

Adapted from: A. Bifet, J.Gama, M. Pechenizkiy, **I.Zliobaite**
Handling Concept Drift: Importance, Challenges and Solutions

Types of changes in data

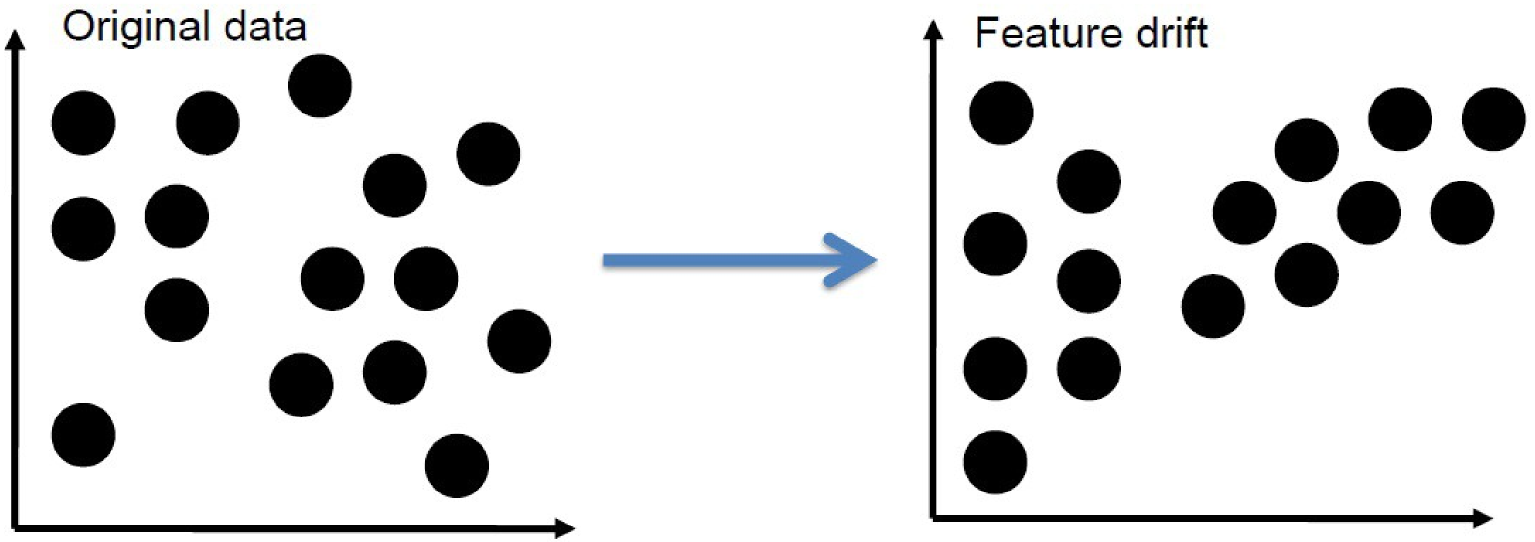
- Evolving/drifting data = data distribution changes over time

- Feature drift [data evolution]
  - distribution of input data X changes, $p(X)$

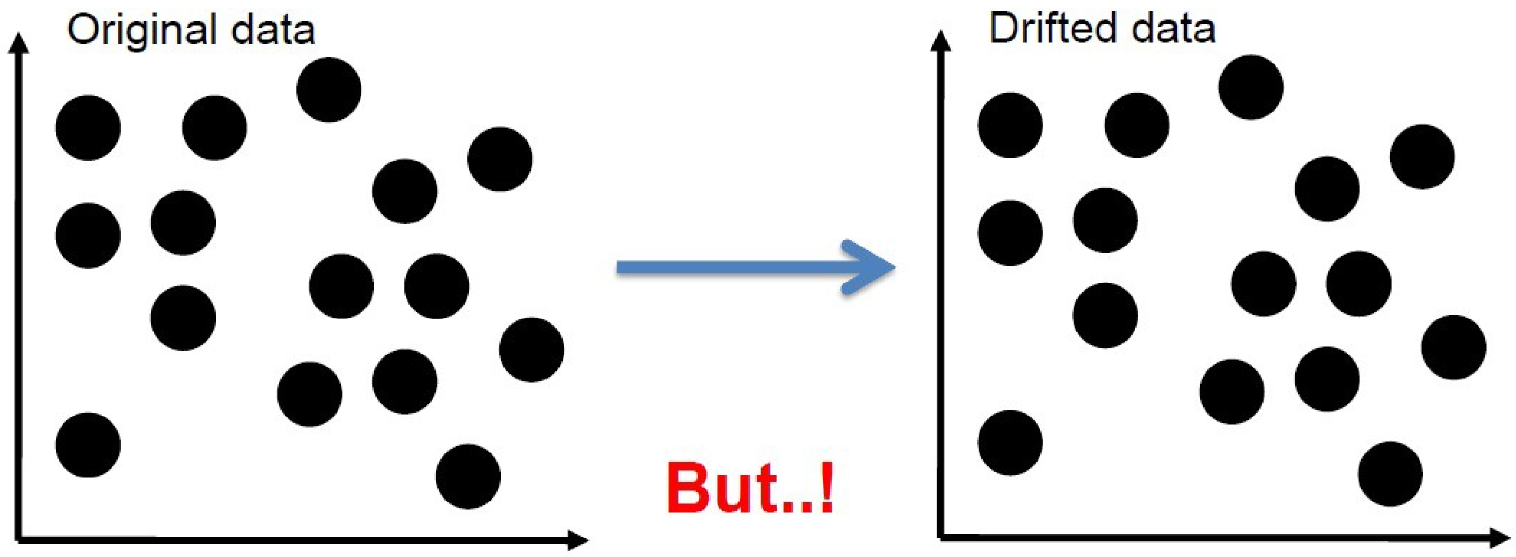
- Real concept drift
  - relation between input X and target y changes, $p(y|X)$

- Changing prior distribution
  - E.g. of the target $p(y)$
  - Arrival of new information new concepts/classes appear

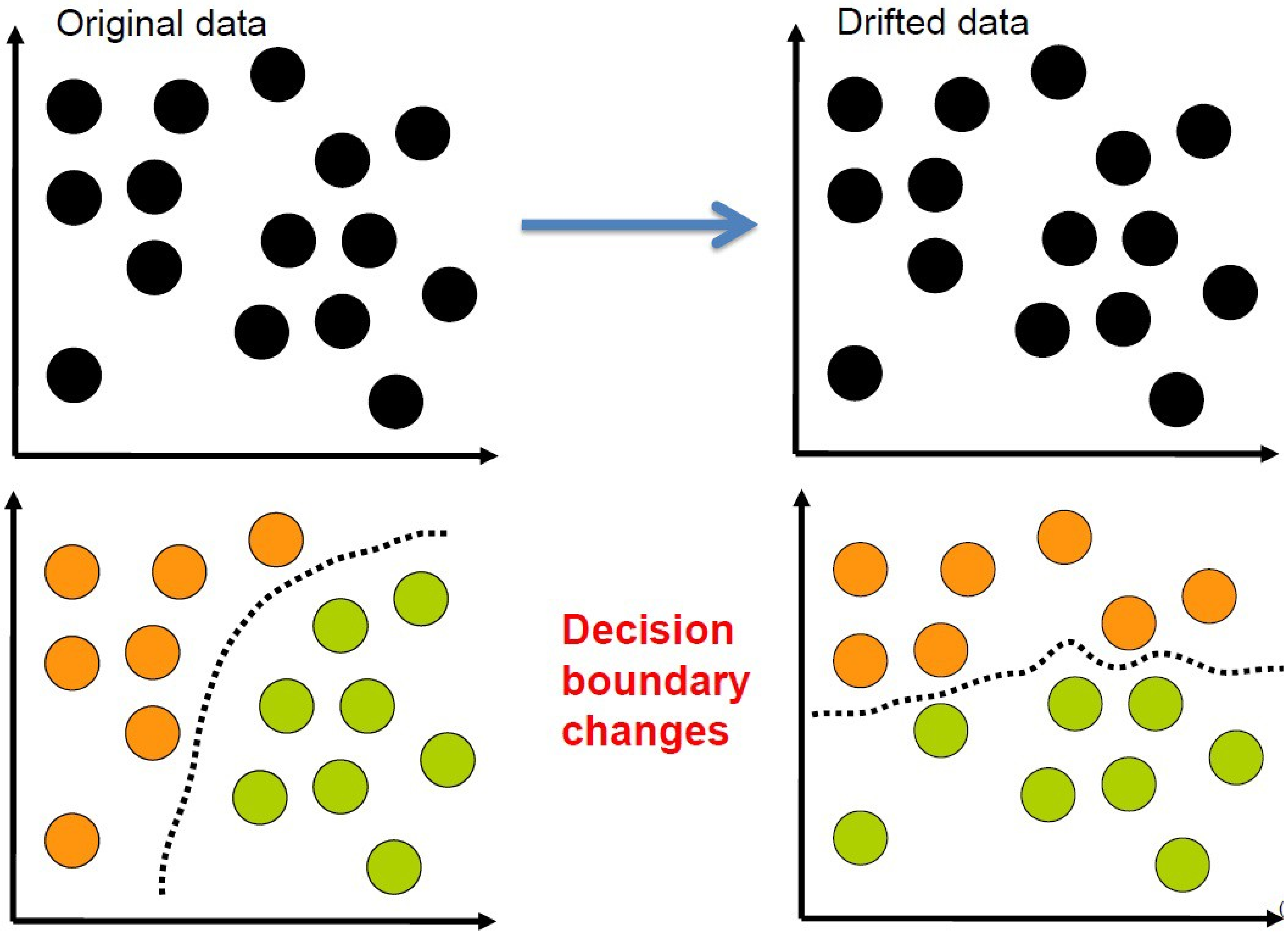
Feature drift [data evolution]



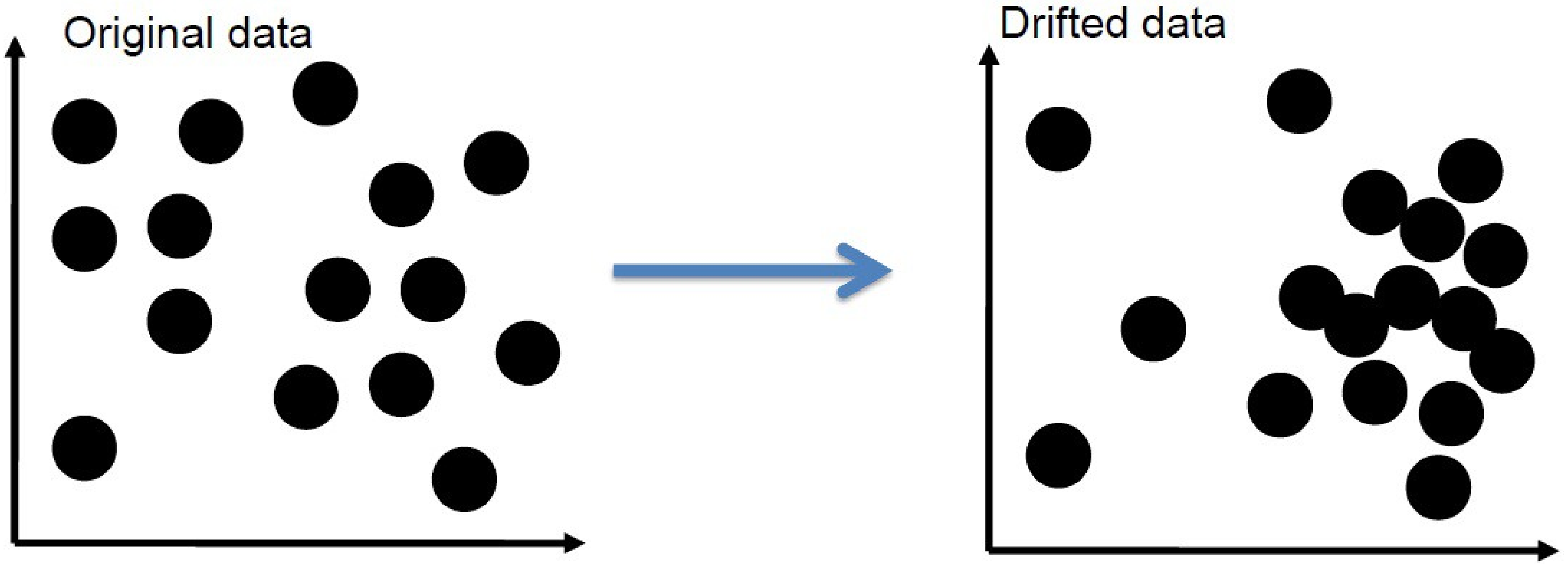
Real concept drift



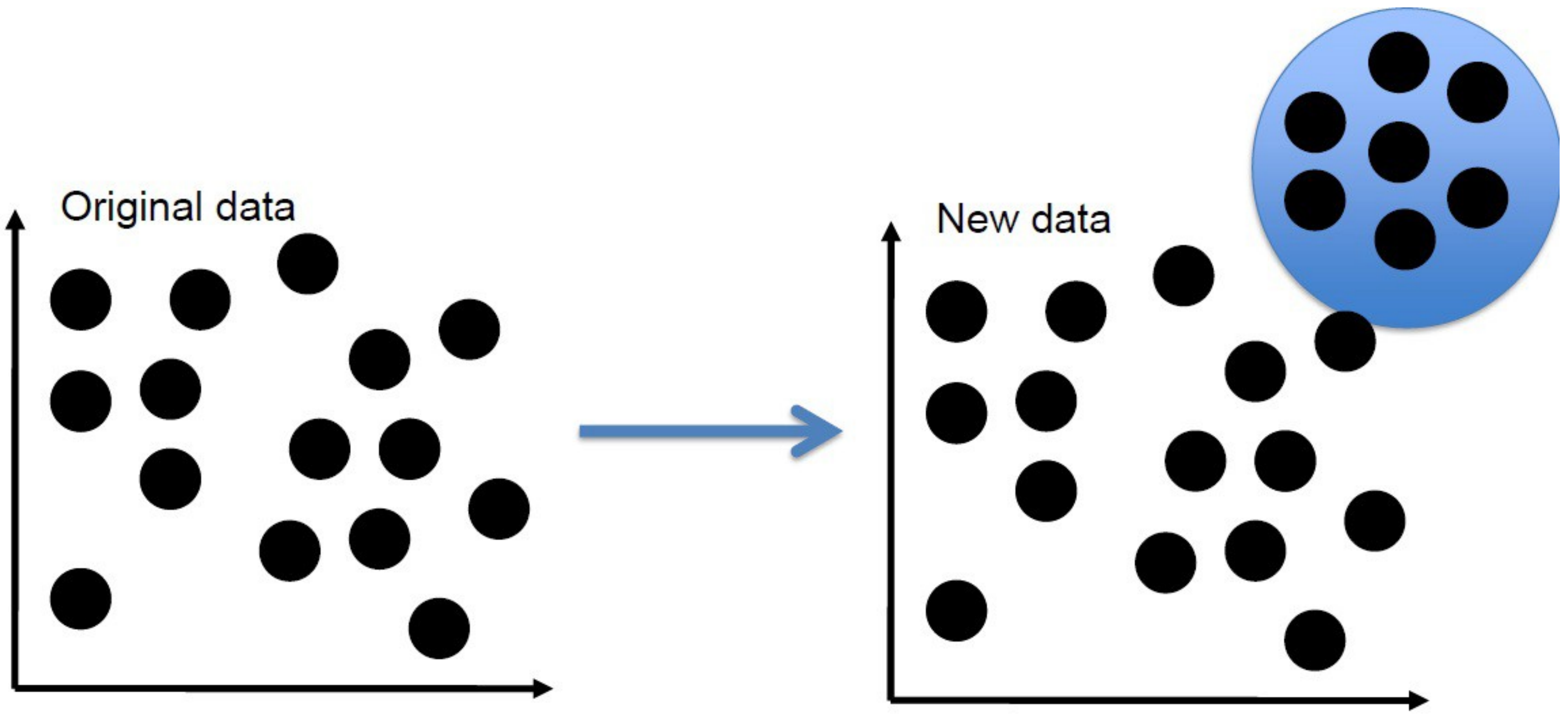
Real concept drift



Changing priors



Arrival of new information



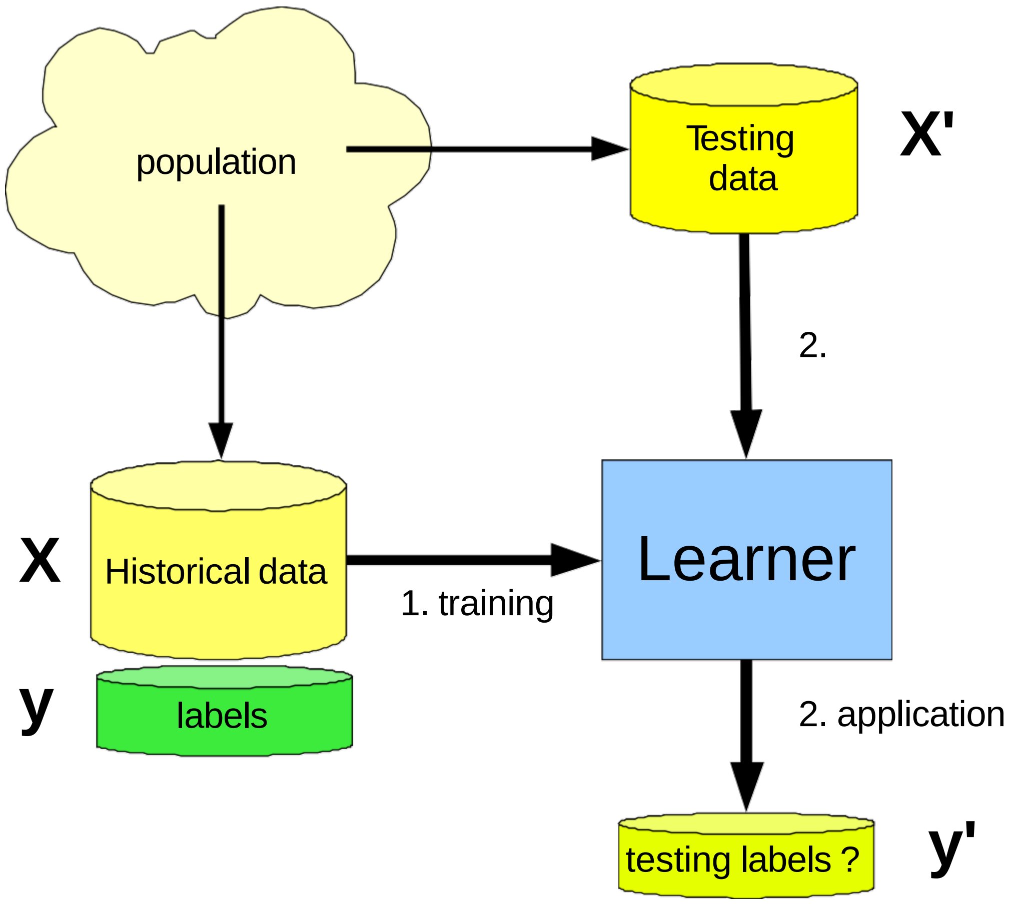
Problem of capturing concept drift / change

Given an input sequence $x_1, x_2, \ldots, x_t$ we want to output at instant t an alarm signal if there is a distribution change and also a prediction $\hat{x}_{t+1}$ minimizing prediction error:

- $||\hat{x}_{t+1} - x_{t+1}||$

- Outputs
  - an estimation of some important parameters of the input distribution, and
  - a signal alarm indicating that distribution change has recently occurred.

Supervised Learning



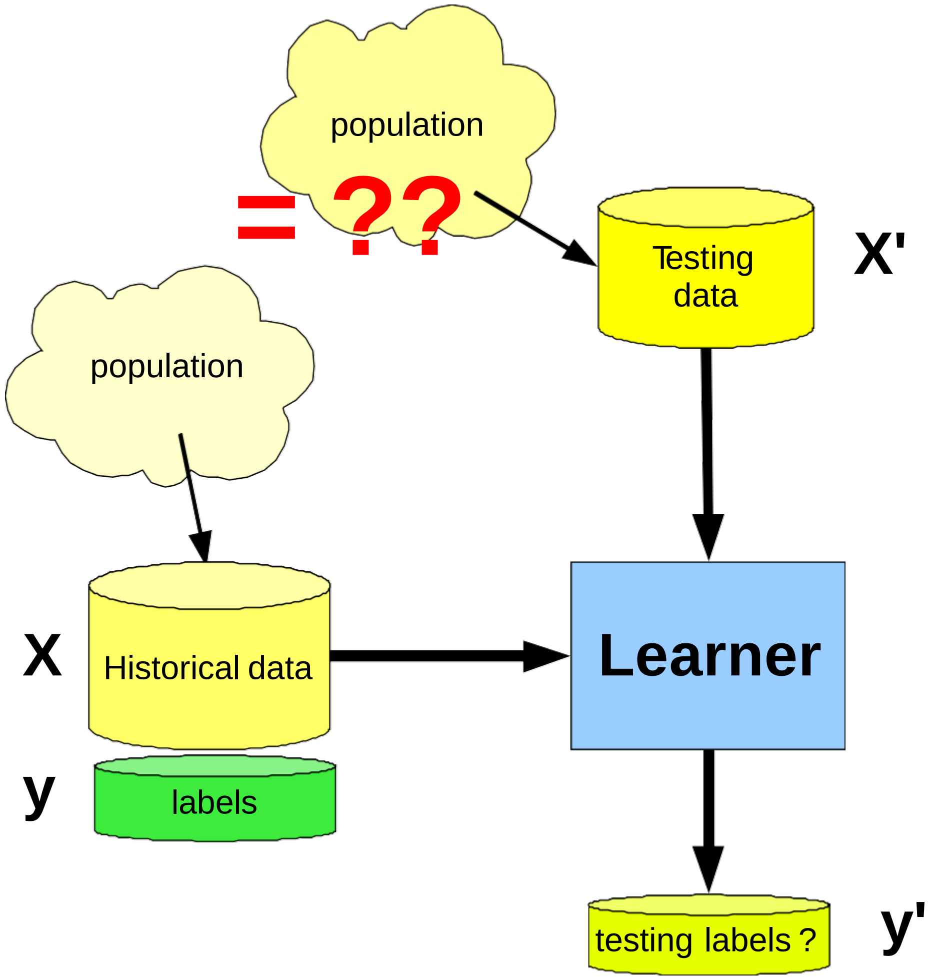
Training:
Learning
a mapping function

$$y = L\,(x)$$

Application:
Applying ***L***
to unseen data

$$y' = L\,(X')$$

Learning with concept drift



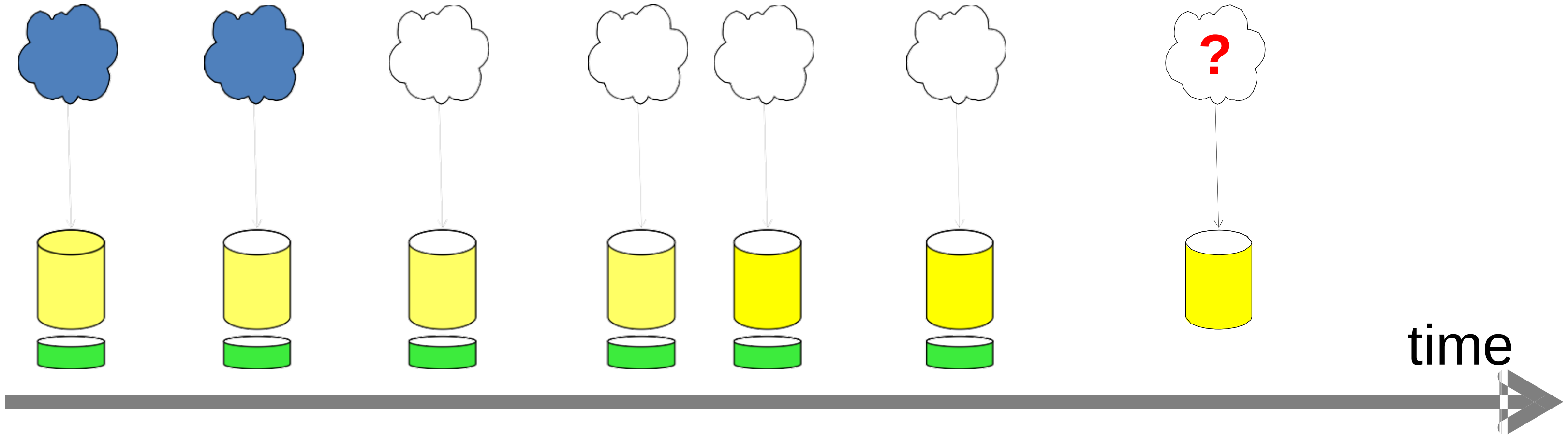
Training:

y = *L* (x)

Application:

y' = *L??* (X')

Online setting

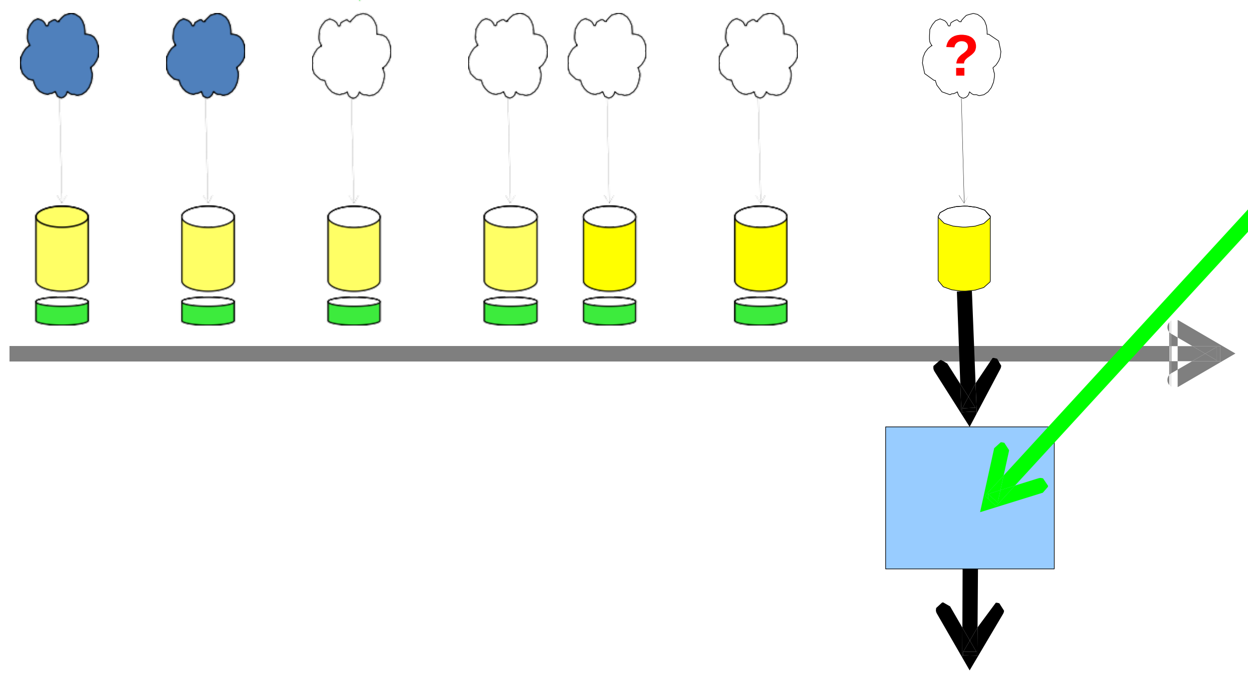


Data arrives in a stream

How to Adapt

Select the right training data and *retrain*

or adjust the model *incrementally*



Desired Properties of a System To Handle Concept Drift

- Adapt to concept drift asap
- Distinguish noise from changes
  - robust to noise, but adaptive to changes
- Recognizing and reacting to reoccurring contexts
- Adapting with limited resources
  - time and memory

Adaptive learning strategies

	Triggering	Evolving
Single classifier	Detectors variable windows	Forgetting fixed windows, Instance weighting
Ensemble	Contextual dynamic integration, meta learning	Dynamic ensemble adaptive combination rules

adaptive learning approaches implicitly or explicitly assume some type of change

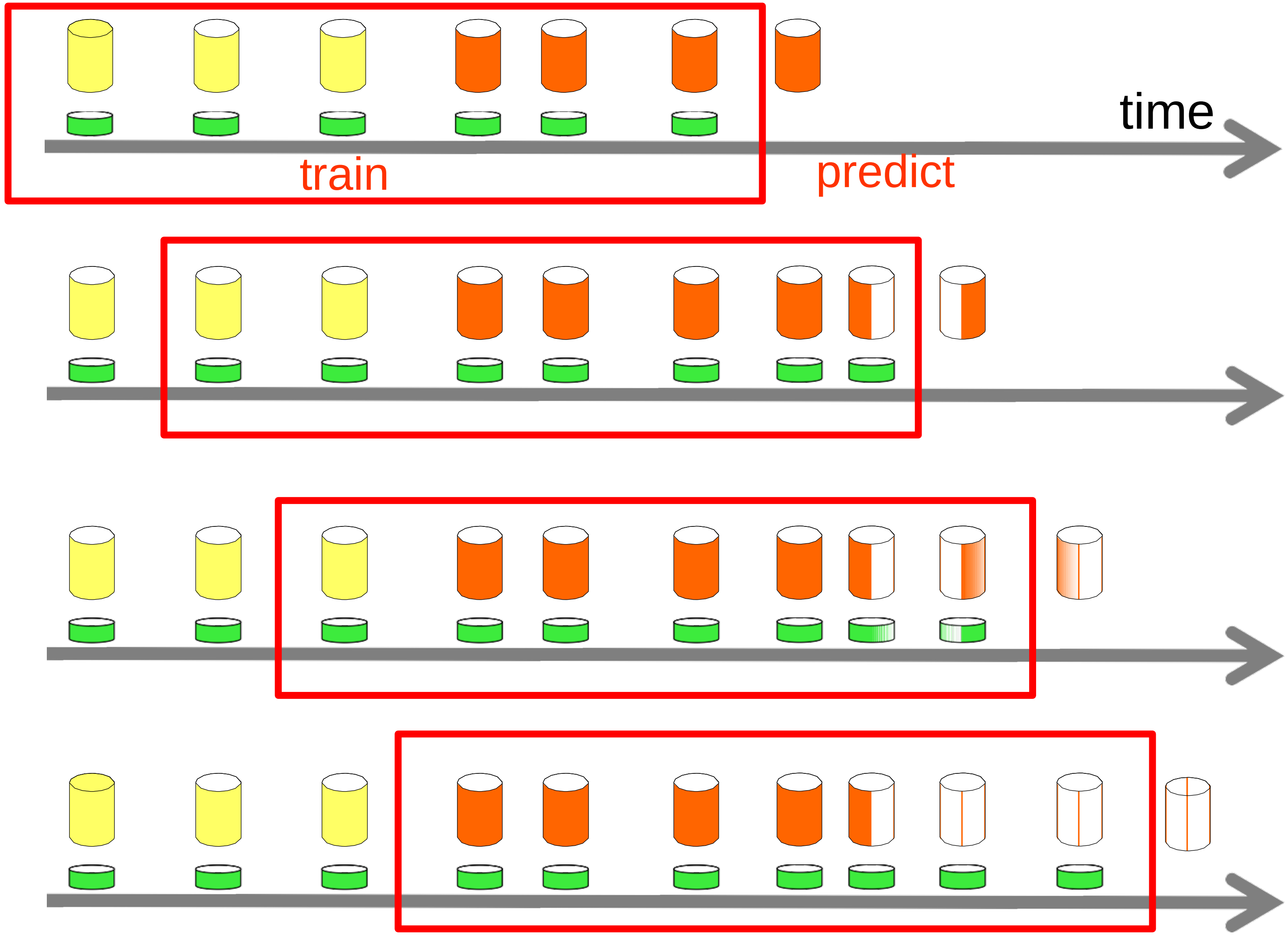
Adaptive learning strategies

	Triggering	Evolving
Single classifier	**Detectors** variable windows	**Forgetting** fixed windows, Instance weighting
Ensemble	**Contextual** dynamic integration, meta learning	**Dynamic ensemble** adaptive combination rules

forget old data and retrain at a fixed rate

Blind adaptation

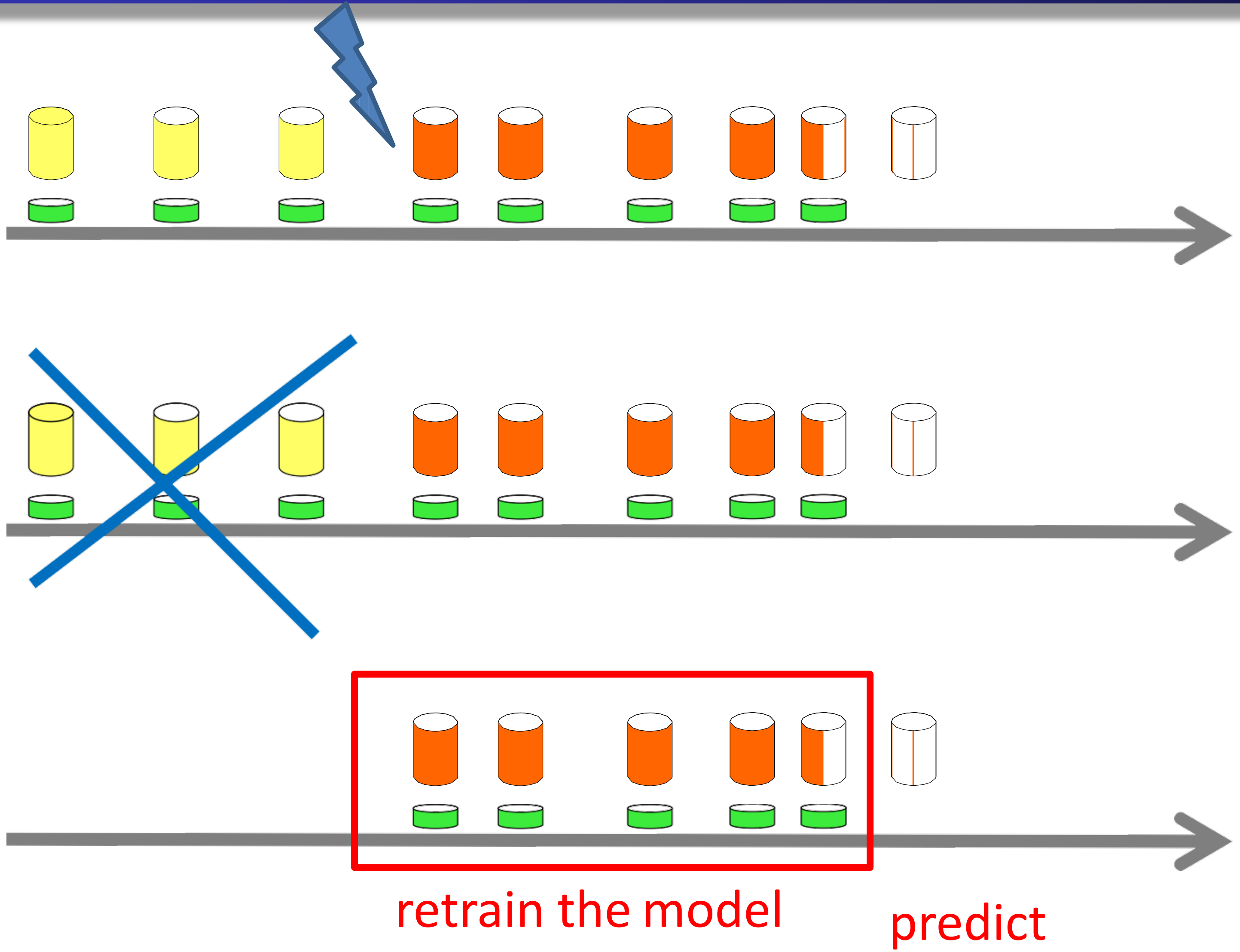
Fixed Training Window



Adaptive learning strategies

	Triggering	Evolving
detect a change and cut **Informed Adaptation** Single classifier	Detectors variable windows	Forgetting fixed windows, Instance weighting
Ensemble	Contextual dynamic integration, meta learning	Dynamic ensemble adaptive combination rules

Variable Training Window

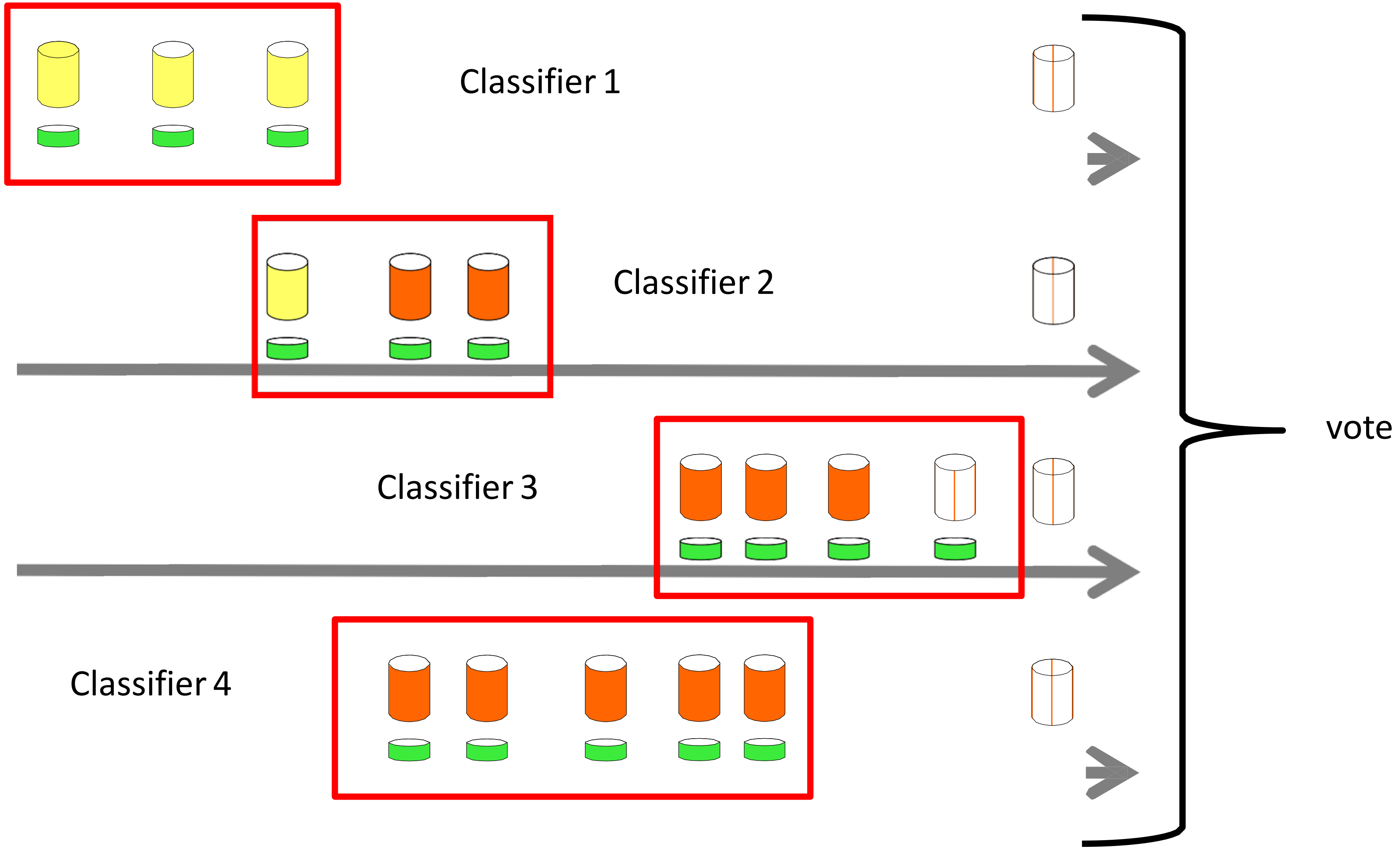


Adaptive learning strategies

	Triggering	Evolving
Single classifier	**Detectors** variable windows	**Forgetting** fixed windows, Instance weighting
Ensemble	**Contextual** dynamic integration, meta learning	**Dynamic ensemble** adaptive combination rules

build many models, dynamically combine

Dynamic Ensemble

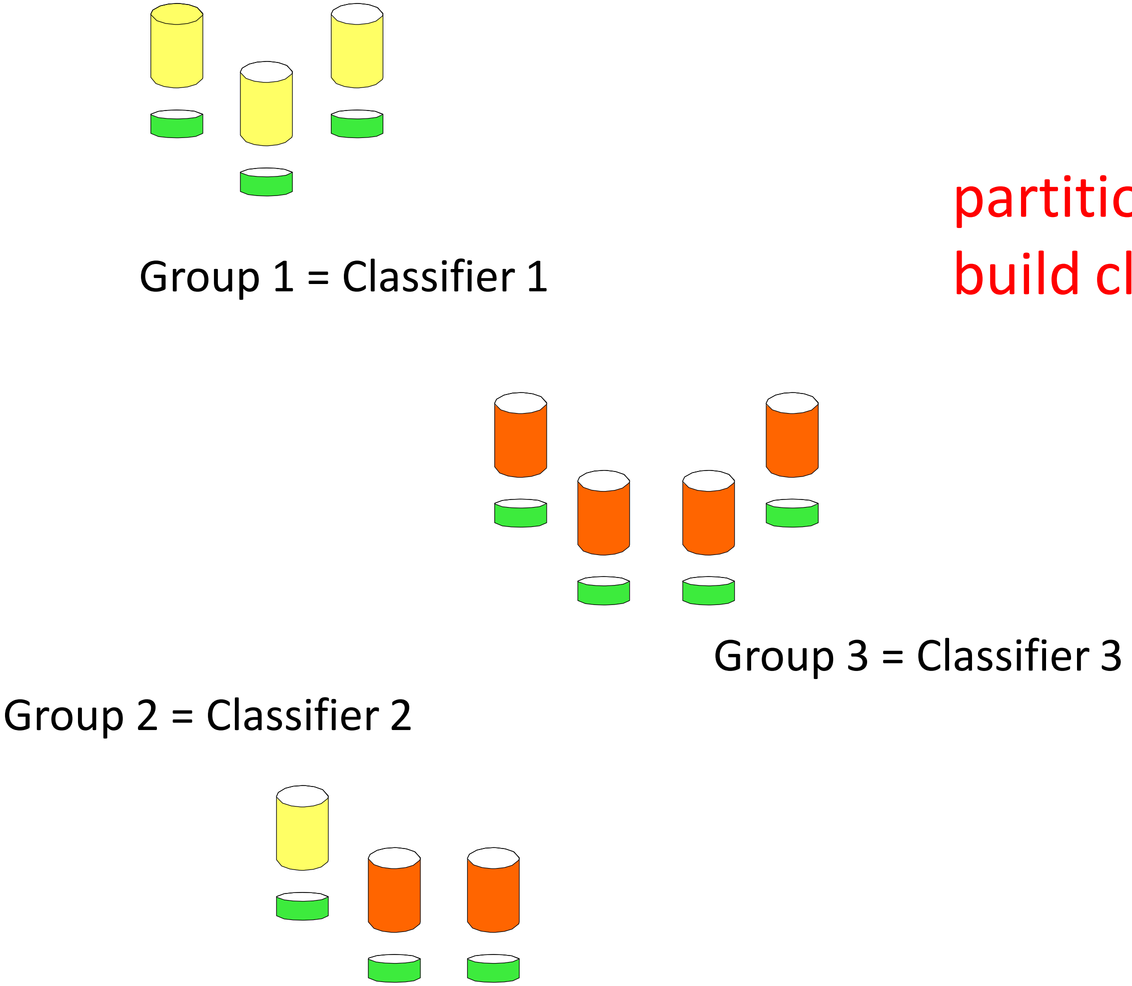


Adaptive learning strategies

	Triggering	Evolving
Single classifier	Detectors variable windows	Forgetting fixed windows, Instance weighting
Ensemble	Contextual dynamic integration, meta learning	Dynamic ensemble adaptive combination rules

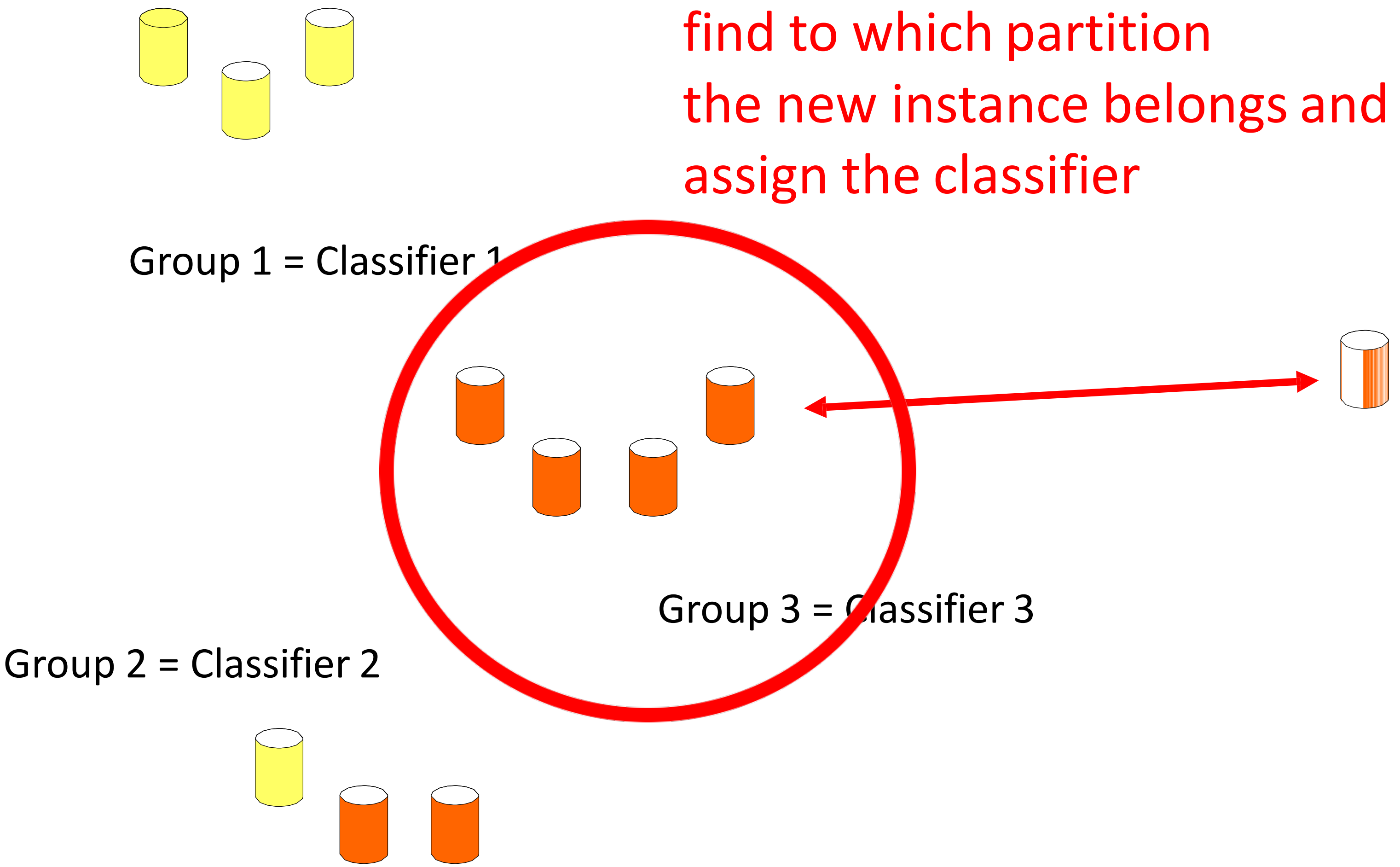
build many models,
switch models according
to the observed incoming data

Contextual (Meta) Approaches

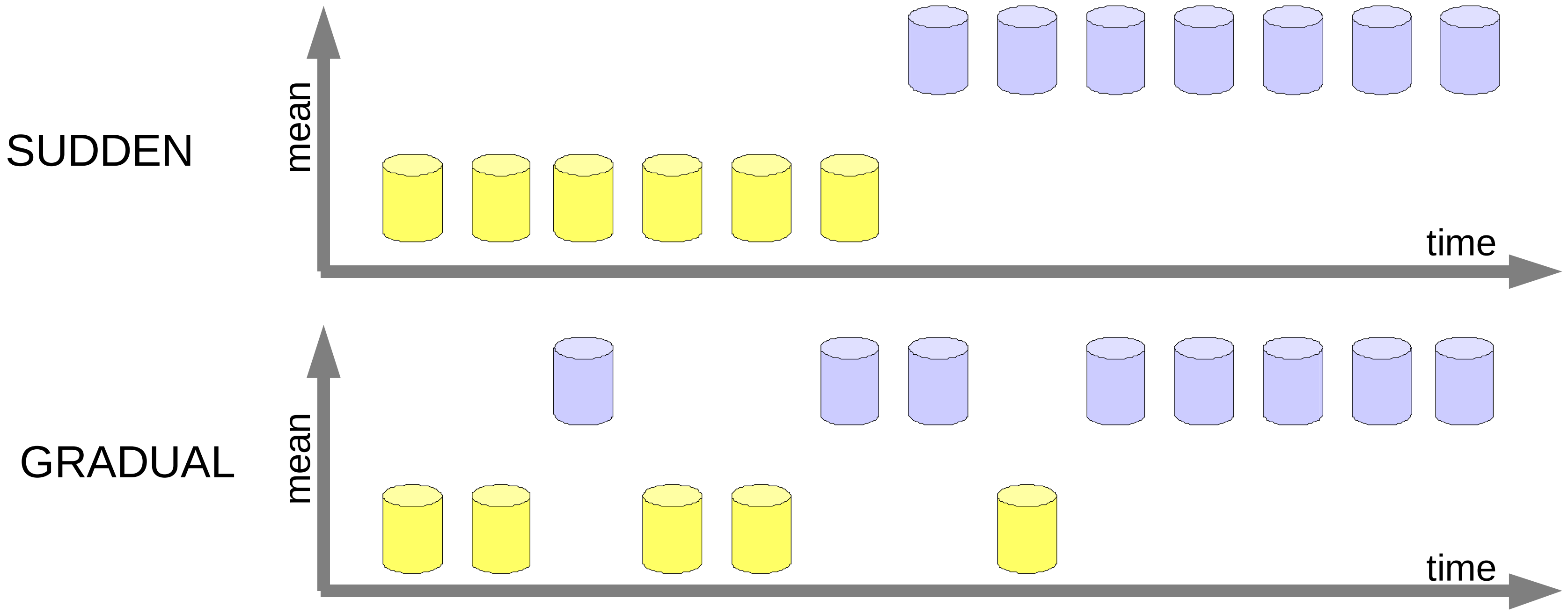


partition the training data
build classifiers

Contextual (Meta) Approaches

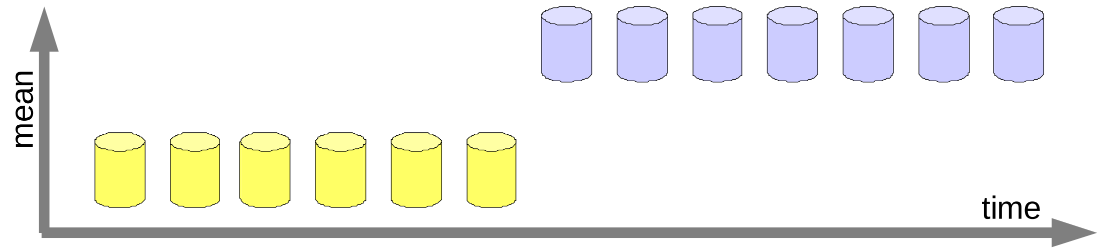


Types of Changes

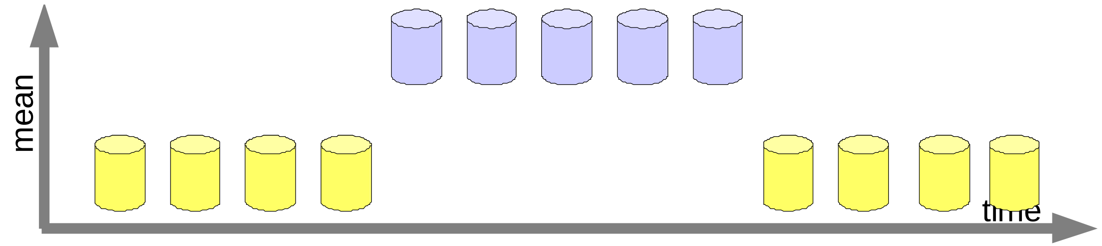


Types of Changes

ALWAYS NEW*



REOCCURING



Adaptive learning strategies

	Triggering	Evolving
Single classifier	Detectors variable windows	Forgetting fixed windows, Instance weighting
Ensemble	Contextual dynamic integration, meta learning	Dynamic ensemble adaptive combination rules

Sudden Drift

Reoccuring Drift

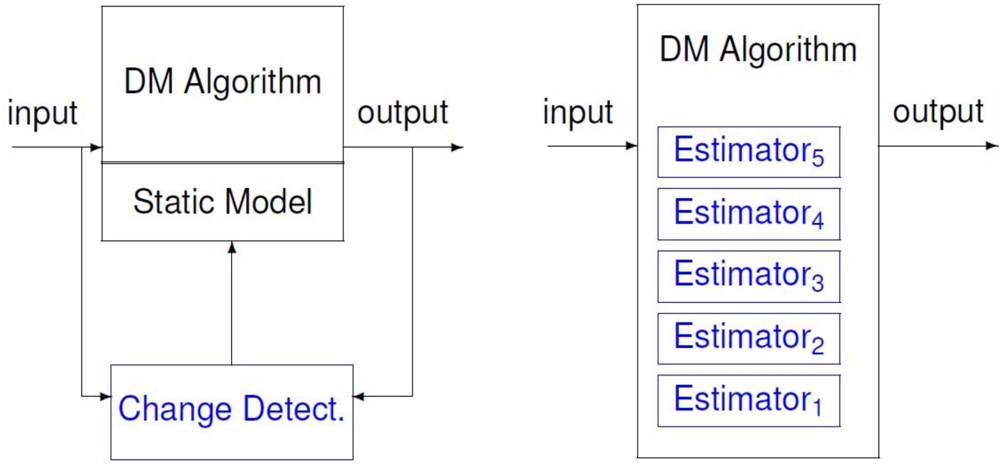
Gradual Drift

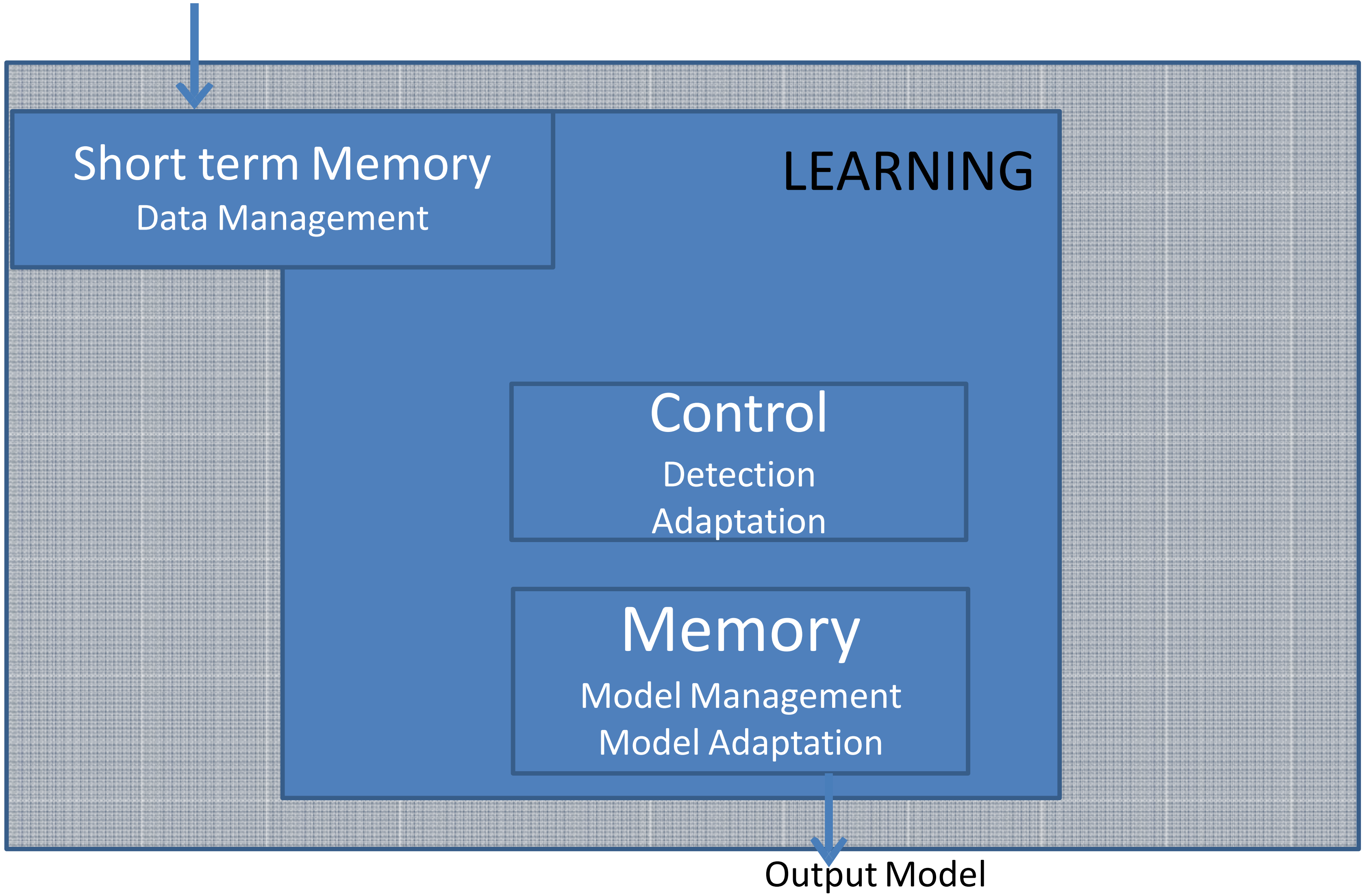
adaptive learning approaches implicitly or explicitly assume some type of change

Which approach to use?

- we need models that evolve over time
- choice of technique depends on
  - what type of change is expected
  - user goals/ applications

- *The goal*: to discuss selected popular approaches from each of the major types in more detail.
  - Data management
  - Detection Methods
  - Adaptation methods
  - Decision model management




Short term Memory
Data Management
LEARNING
Control
Detection
Adaptation
Memory
Model Management
Model Adaptation
Output Model

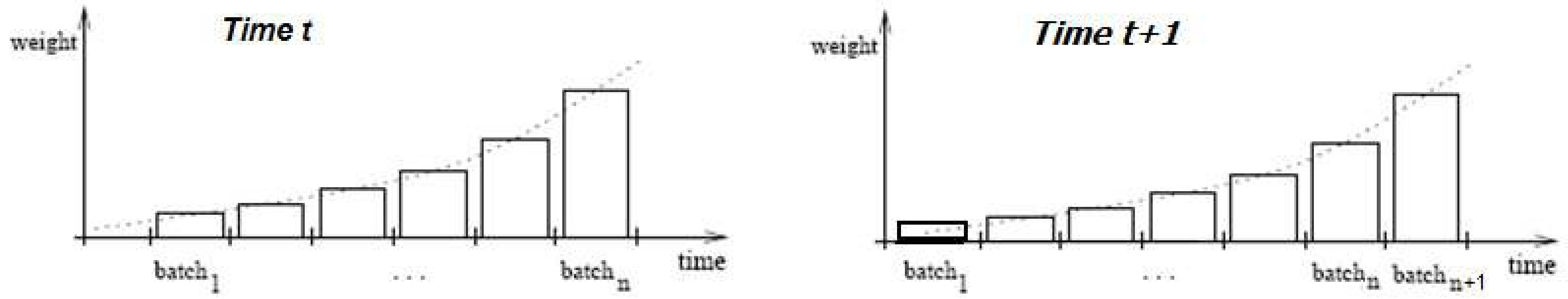
Data Management

- Characterize the information stored in memory to maintain a decision model consistent with the actual state of the nature.
  - Full Memory.
  - Partial Memory.

Weighting examples

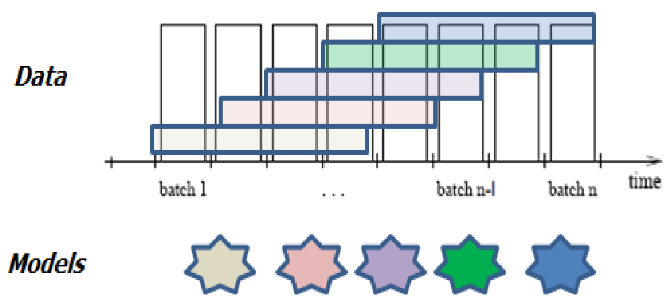
- Full Memory.
  - Store in memory sufficient statistics over all the examples.
  - Weighting the examples accordingly to their age.
  - Oldest examples are less important.
- Weighting examples based on the age:
  - $w_{\lambda}(x) = \exp(-\lambda t_x)$

Where example **x** was found t_x time steps ago



Partial Memory

- Partial Memory.
  - Store in memory only the most recent examples.
  - Examples are stored in a *first-in first-out* data structure.
  - At each time step the learner induces a decision model using only the examples that are included in the window.



- The key difficulty is how to select the appropriate window size:
  - Small window
    - can assure a fast adaptability in phases with concept changes
    - In more stable phases it can affect the learner performance
  - Large window
    - produce good and stable learning results in stable phases
    - can not react quickly to concept changes.

Partial Memory - Windows

- Fixed Size windows
  - Store in memory a fixed number of the most recent examples.
  - Whenever a new example is available:
    - it is stored in memory and
    - the oldest one is discarded.
  - This is the simplest method to deal with concept drift and can be used as a baseline for comparisons.

- Adaptive Size windows
  - the set of examples in the window is variable.
  - They are used in conjunction with a detection model.
  - Decreasing the size of the window whenever the detection model signals drift and increasing otherwise.

Detection Methods

- These methods are ***blind*** adaptation models:
  - There is no explicit change detection
  - Do not provide:
    - » Any indication about change points
    - » Dynamics of the process generating data

Detection Methods

- The Detection Model characterizes the techniques and mechanisms for explicit drift detection.
  - An advantage of the detection model is they can provide:
    - Meaningful descriptions:
      - indicating change-points
      - small time-windows where the change occurs
    - Quantify the changes.

- Monitoring the evolution of performance indicators.
  - Performance measures (Accuracy, recall and precision),
  - Properties of the data, etc
- Monitoring distributions on two different time-windows
  - A reference window over past examples
  - A window over the most recent examples

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.

Concept Drift Detection Techniques

- 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
  - Main Problems:
    - How to distinguish Change from Noise?
    - How to React to drift?

The CUSUM Test

- Cumulative sum algorithm (CUSUM).
- The cumulative sum (CUSUM algorithm), gives an alarm when the mean of the input data is significantly different from zero.
- The CUSUM test is memoryless, and its accuracy depends on the choice of parameters λ and α.

Detecting significant increases:

  - **$g_0 = 0$**
  - **$g_t = \max(0; g_{t-1} + (x_t - \alpha))$**

- The decision rule is:
  - **if $g_t > \lambda$ then alarm and $g_t = 0$.**

Detecting decreases:

  - **$g_0 = 0$**
  - **$g_t = \min(0; g_{t-1} + (x_t - \alpha))$**

- The decision rule is:
  - **if $g_t < \lambda$ then alarm and $g_t = 0$.**

The CUSUM test is used to detect significant increases (or decreases) in the successive observations of a random variable **x**

Page-Hinckley Test

- The PH test is a sequential adaptation of the detection of an abrupt change in the average of a Gaussian signal.
  - It considers a cumulative variable m_T, defined as the cumulated difference between the observed values and their mean till the current moment:

$$m_{t+1} = \sum_{1}^{t} (x_t - \bar{x}_t + \alpha)$$

- where $\bar{x} = 1/t \sum_{l=1}^{t} x_l$ and
- α corresponds to the magnitude of changes that are allowed.

Page-Hinckley Test

$$m_{t+1} = \sum_{1}^{t} (x_t - \bar{x}_t + \alpha)$$

- where $\bar{x} = 1/t \sum_{l=1}^{t} x_l$ and
- α corresponds to the magnitude of changes that are allowed.

- The minimum value of $\mathbf{m_t}$ is also computed:

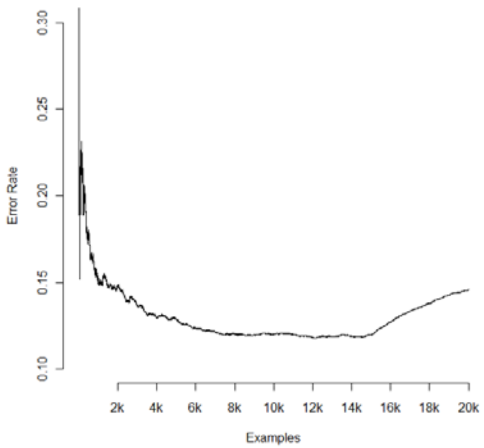
$$M_T = \min(m_t \; ; t = 1, \ldots, T).$$

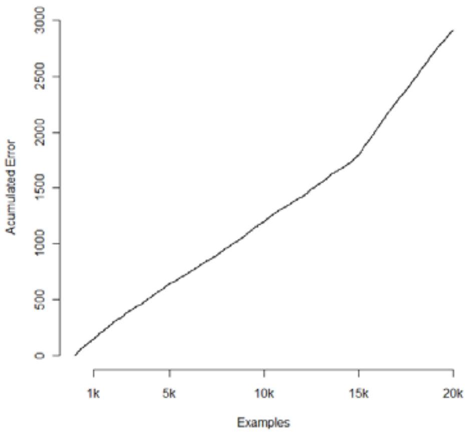
- The test monitors the difference between M_T and m_T :

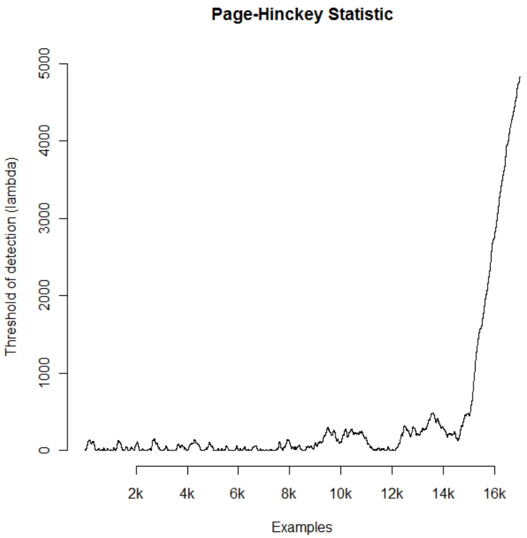
$$PH_T = m_T - M_T \; .$$

- When this difference is greater than a given threshold (λ) we alarm a change in the distribution.

Page-Hinckley Test







Page-Hinckley Test

- The PH_t test is memoryless, and its accuracy depends on the choice of parameters α and λ.
  - Both parameters are relevant to control the trade- off between earlier detecting true alarms by allowing some false alarms.
- The threshold λ depends on the admissible false alarm rate.
  - Increasing λ will entail fewer false alarms, but might miss some changes.

SPC-Statistical Process Control (or DDM)

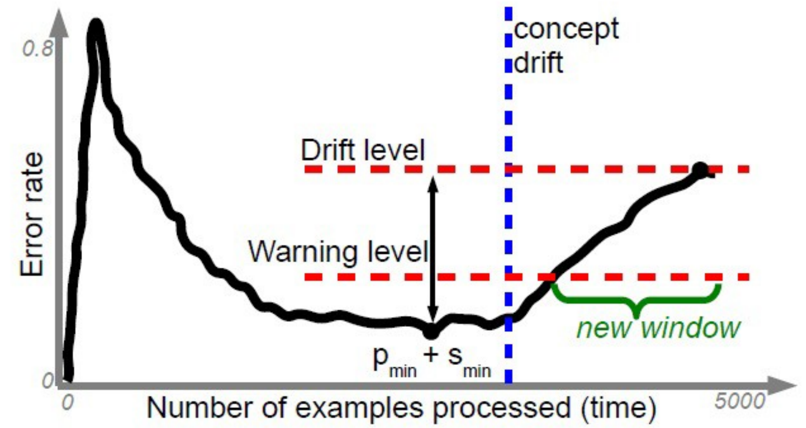
Learning with Drift Detection ,Gama, Medas, Gladys, Rodrigues; SBIA- LNCS Springer, 2004.

- Suppose a sequence of examples in the form < $\mathbf{x}_i$,y_i>
- The actual decision model classifies each example in the sequence
  - In the 0-1 loss function, predictions are either True or False
  - The predictions of the learning algorithm are sequences: T, F, T, F, T, F, T, T, T, F, …
  - The Error is a random variable from Bernoulli trials.
  - The Binomial distribution gives the general form of the probability of observing a F: $p_i = (\#F/i)$

$$s_i = \sqrt{p_i(1 - p_i)/i}$$ where *i* is the number of trials.

SPC-Statistical Process Control

- Maintains two registers:
  - P_{min} and S_{min} such that $P_{min} + S_{min} = \min(p_i + s_i \;)$
  - Minimum of the error rate taking into account the variance of the estimator.
- At example j : the error of the learning algorithm will be
  - Out-control if $p_j + s_j > p_{min} + \alpha \;\; s_{min}$
  - In-control if $p_j + s_j < p_{min} + \beta \;\; s_{min}$
  - Warning Level if $p_{min} + \alpha \;\; s_{min} > p_j + s_j > p_{min} + \;\; s_{min}$
- The constants α and β depend on the desired confidence level.
  - Admissible values are $\alpha = 2$ and $\beta = 3$.



ADWIN: Adaptive Data Stream Sliding Window

Learning from Time-Changing Data with Adaptive Windowing, A.Bifet, R.Gavaldà (SDM'07)

- Whenever two "large enough" subwindows of W exhibit "distinct enough" averages,
  - we can conclude that the corresponding expected values are different, and
  - the older portion of the window is dropped

Let W = $\boxed{101010110111111}$

1	01010110111111
1010	10110111111
1010101	10111111
1010101101	11111
10101011011111	1

ADWIN0: ADAPTIVE WINDOWING ALGORITHM

```
Initialize Window W
for each t > 0
      do W ← W ∪ {x_t} (i.e., add x_t to the head of W)
         repeat Drop elements from the tail of W
           until |μ̂_{W_0} − μ̂_{W_1}| < ε_cut holds
              for every split of W into W = W_0 · W_1
         output μ̂_W
```

Hoeffding Bound

Exponential Histogram

$M = 2$

	1010101	101	11	1	1	1
Content:	4	2	2	1	1	1
Capacity:	7	3	2	1	1	1

	1010101	101	11	11	1
Content:	4	2	2	2	1
Capacity:	7	3	2	2	1

	1010101	10111	11	1
Content:	4	4	2	1
Capacity:	7	5	2	1

SEED

- 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.

$$W = \boxed{B_1 \mid B_2 \; B_3 \; B_4 \; B_5}$$
$$W = \boxed{B_1 \; B_2 \mid B_3 \; B_4 \; B_5}$$
$$W = \boxed{B_1 \; B_2 \; B_3 \mid B_4 \; B_5}$$
$$W = \boxed{B_1 \; B_2 \; B_3 \; B_4 \mid B_5}$$

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

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

Concept Drift Evaluation

- Mean Time between False Alarms (MTFA)
- Mean Time to Detection (MTD)
- Missed Detection Rate (MDR)
- Average Run Length (ARL(ϑ))

The design of a change detector is a compromise between detecting true changes and avoiding false alarms.

Concept Drift Evaluation

Main properties of an optimal change detector and predictor system.

- High accuracy in the prediction
- Low mean time to detection (MTD), false positive rate (FAR) and missed detection rate (MDR)
- Low computational cost: minimum space and time needed
- Theoretical guarantees
- No parameters needed

Scenario

- The characteristics of the data stream you are analysing:
  - Abrupt concept drifts which happens infrequently.
- Constraints:
  - You are analysing this in a memory resource constrained environment.

Scenario

- The characteristics of the data stream you are analysing:
  - No information on this is given.
- Constraints:
  - The timeliness of dealing with dealing with the instances is an issue, all instances needs to be dealt with in an efficient manner.
  - Low false alarm and maintaining a good/high TP rate.