Practical Machine Learning for Streaming Data

ACM SIGKDD Tutorial (hands-on)
2024

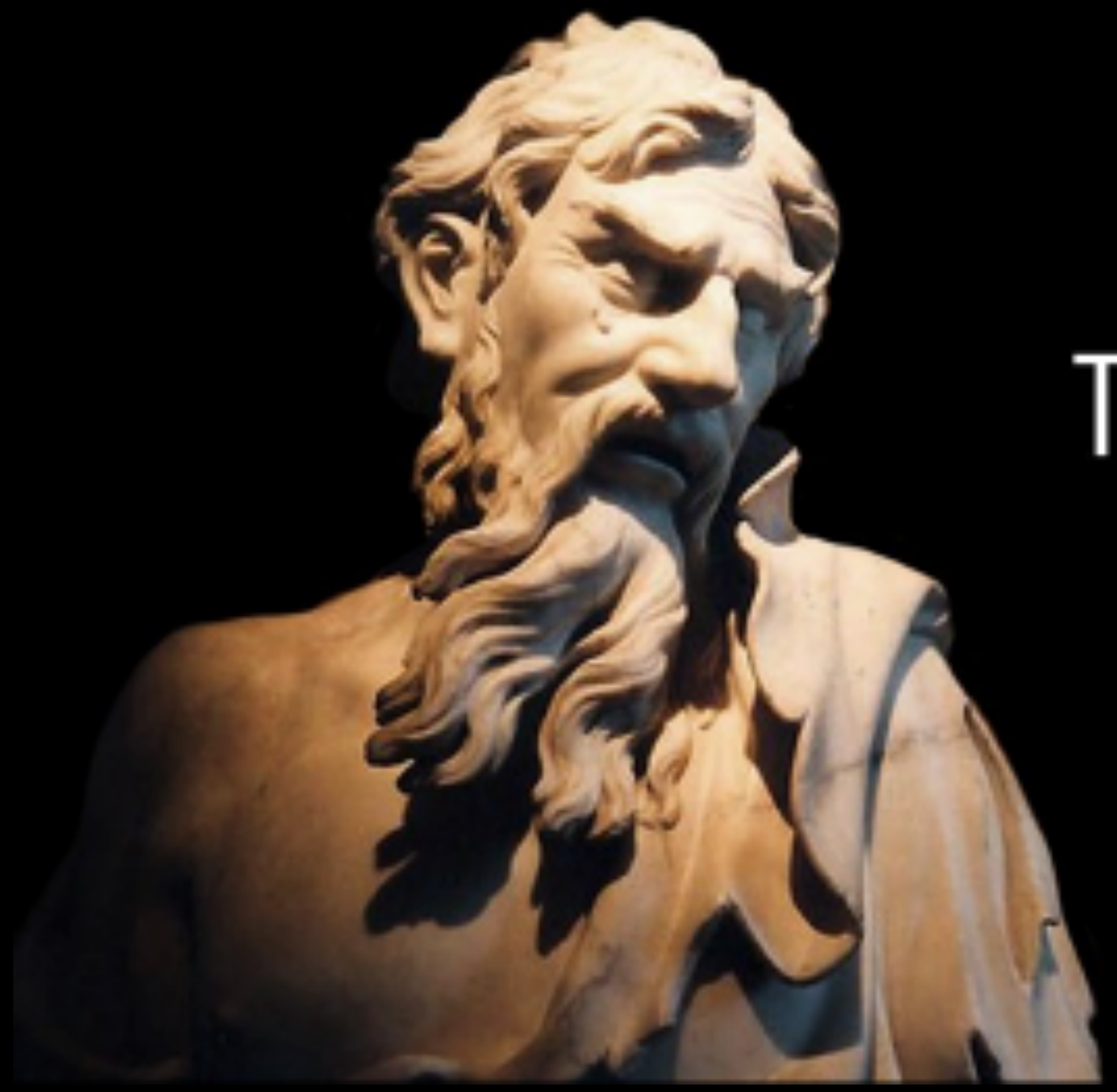
Heitor Murilo Gomes[1], Albert Bifet[2,3]

https://adaptive-machine-learning.github.io/kdd2024_ml_for_streams/

[1] Victoria University of Wellington, New Zealand, [2] University of Waikato, New Zealand,
[3] TELECOM Paris, LCTI, France.

Concept Drifts

Evolving Stream Learning



There is nothing **permanent** except **change.**

– *Heraclitus*

Evolving Stream Learning

- The world is dynamic… **changes occur all the time**

- These **changes affect** our **machine learning models**

Evolving Stream Learning

- The world is dynamic… **changes occur all the time**
- These **changes affect** our **machine learning models**

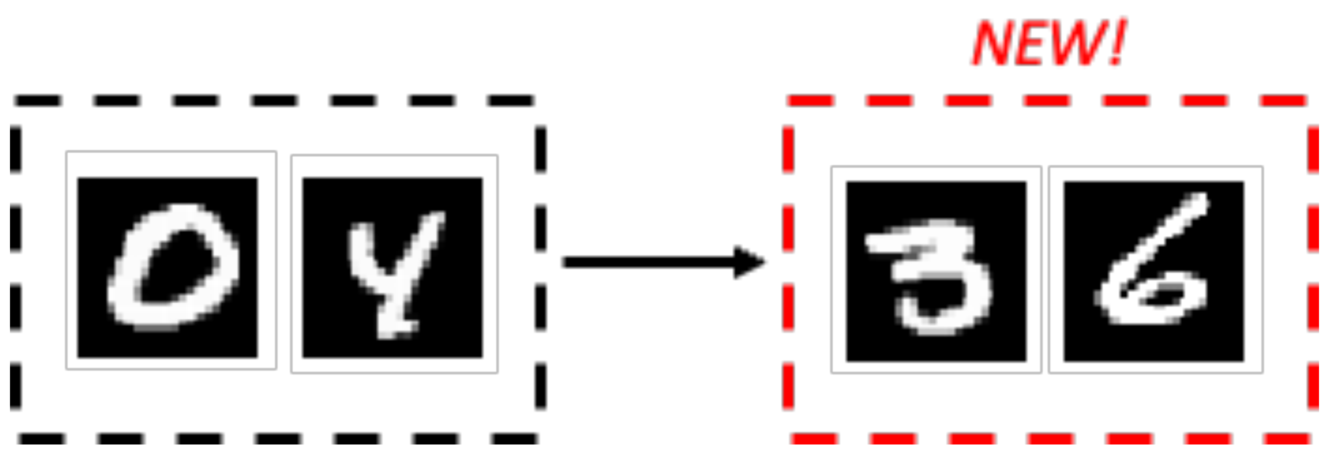
Ideally, we would like to…

(1) **Detect, understand** and **react to changes** in the data

(2) **Learn new concepts** without **forgetting old concepts**

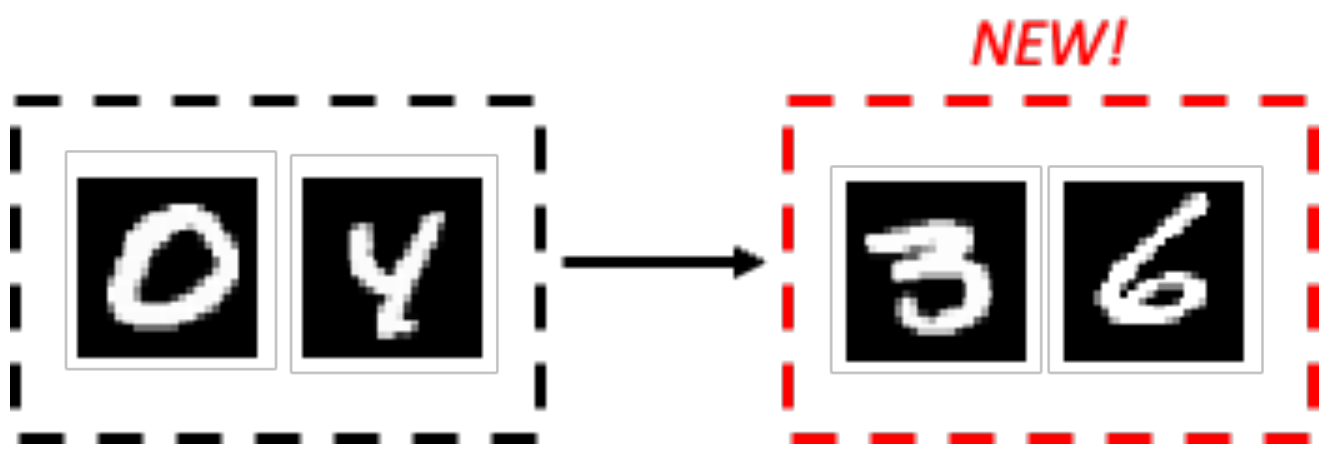
Some Examples

Learn to classify **new classes**

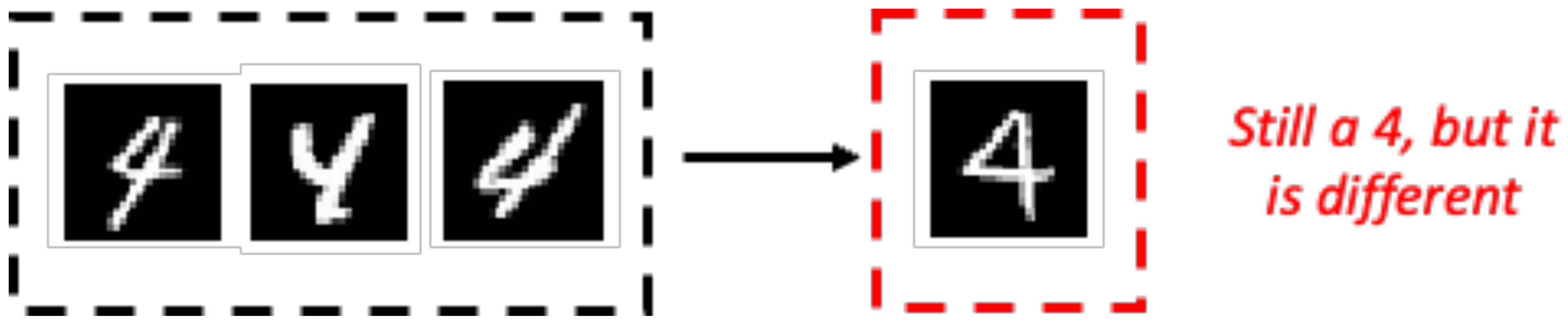


Some Examples

Learn to classify **new classes**

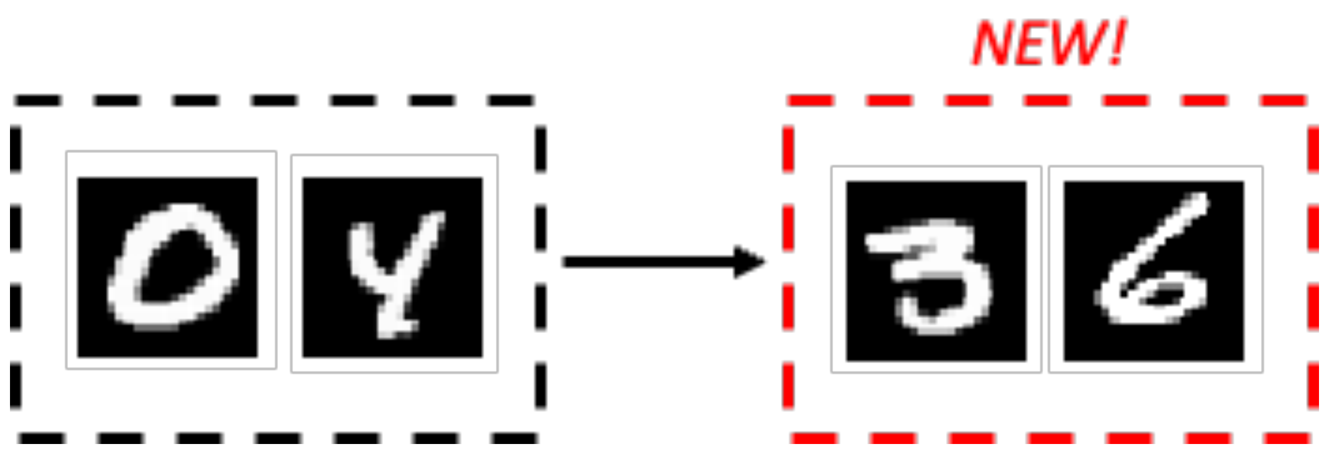


Update itself to accommodate for **changes within existing classes**

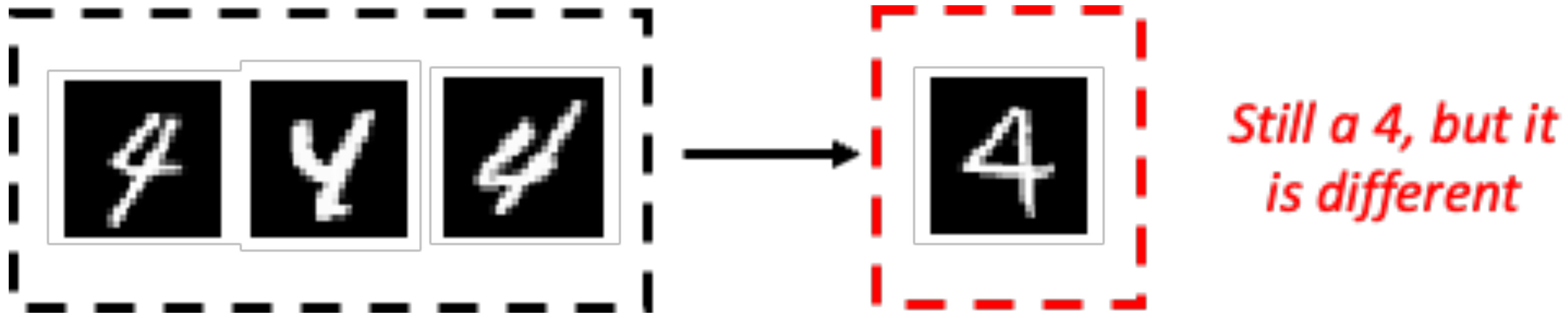


Some Examples

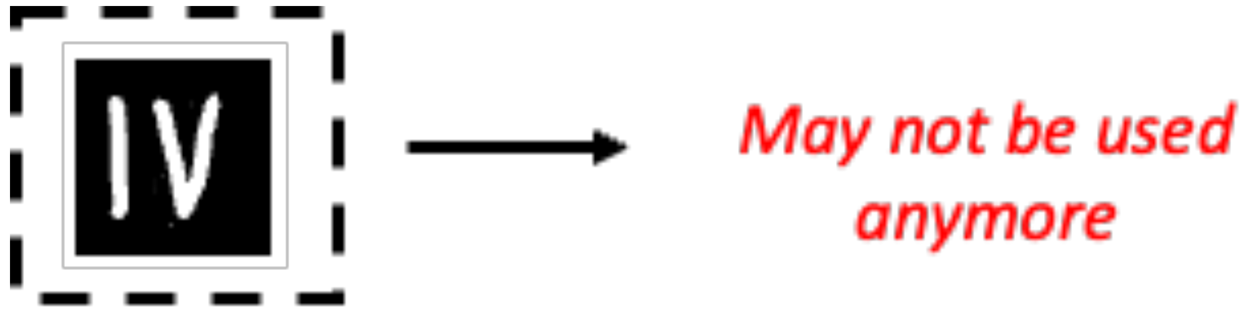
Learn to classify **new classes**



Update itself to accommodate for **changes within existing classes**

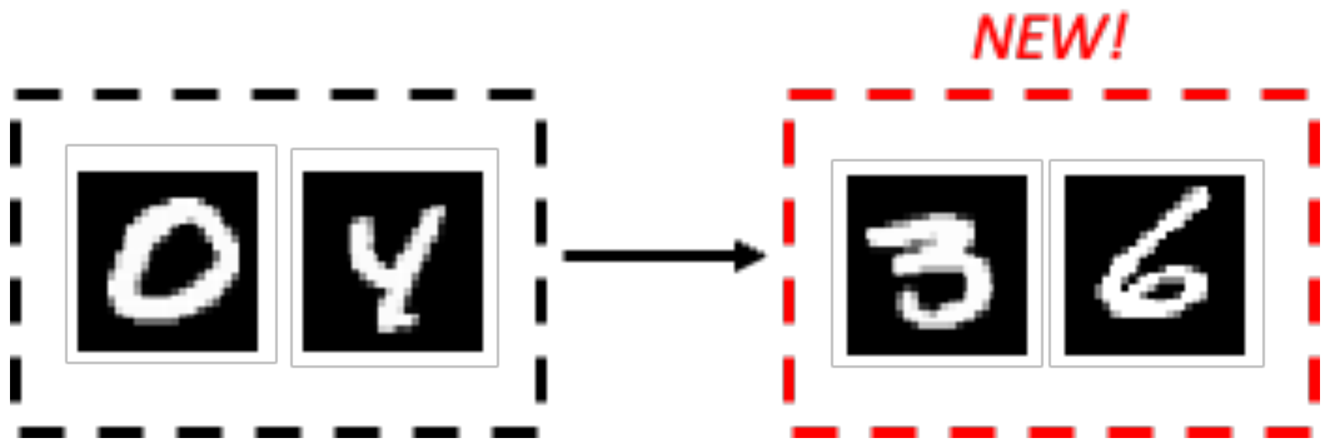


Forgets that which is **no longer needed**



Some Examples

Learn to classify **new classes**

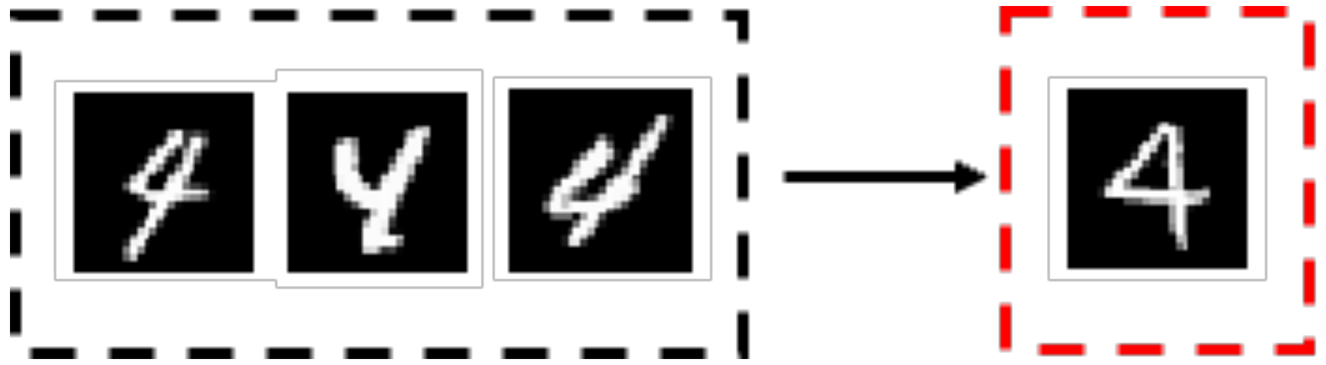


Related Research Areas / Jargon

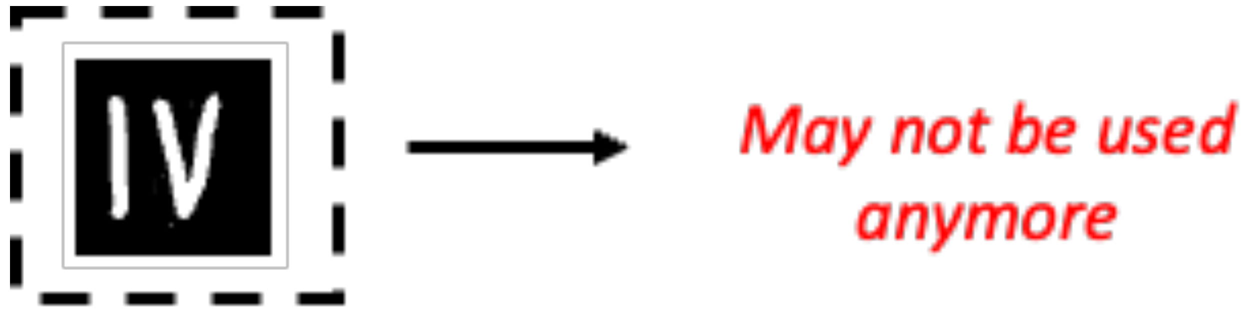
Class Evolution (Stream Learning)

Class Incremental (Continual Learning)

Update itself to accommodate for **changes within existing classes**



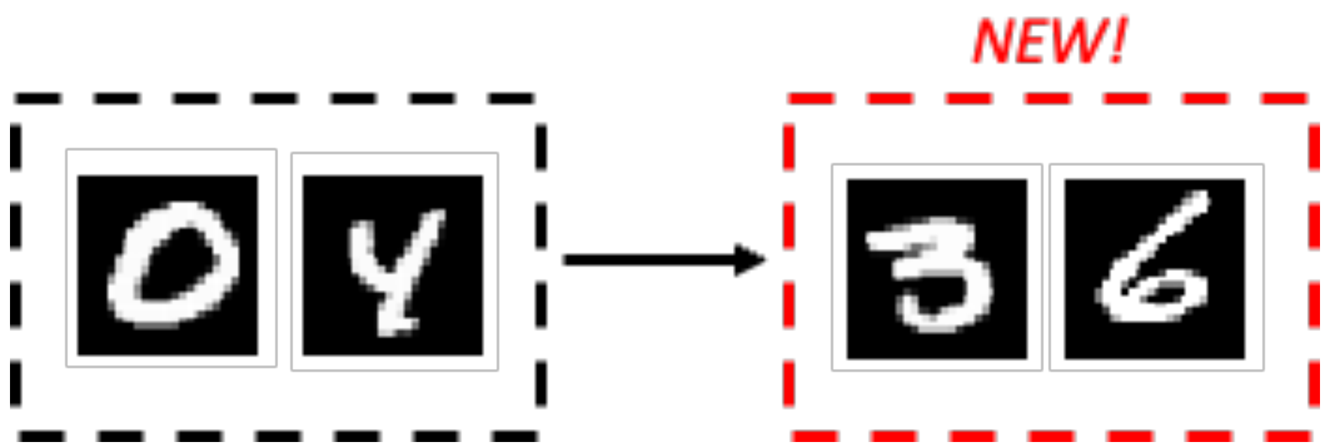
Forgets that which is **no longer needed**



Some Examples

Learn to classify **new classes**

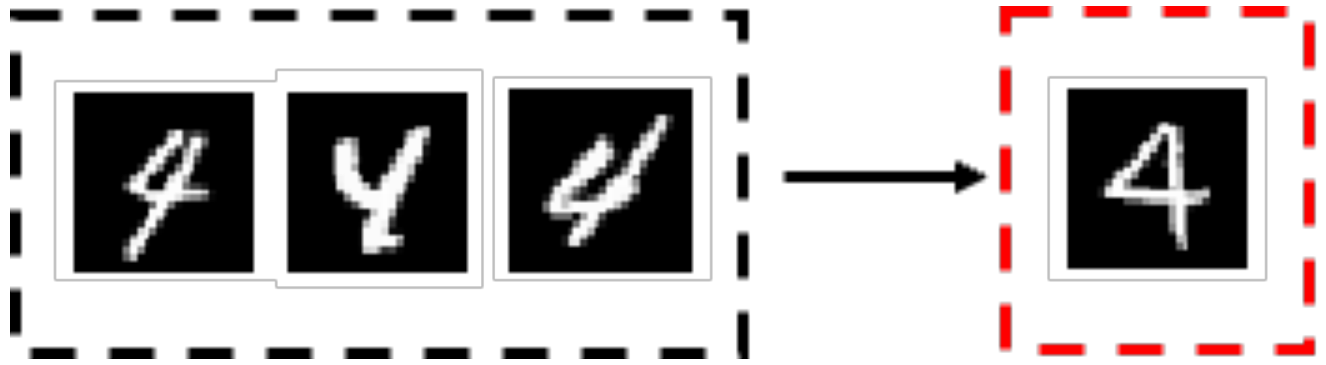
Related Research Areas / Jargon



Class Evolution (Stream Learning)

Class Incremental (Continual Learning)

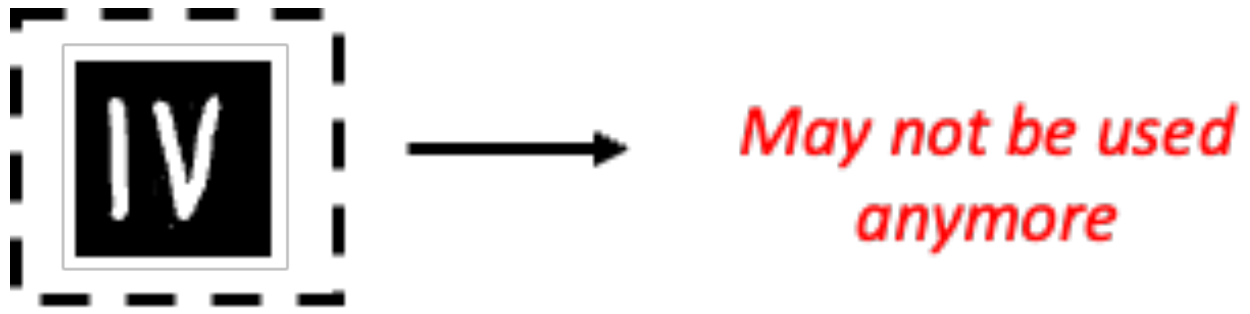
Update itself to accommodate for **changes within existing classes**



Concept Drift (Stream Learning)

Domain Incremental (Continual Learning)

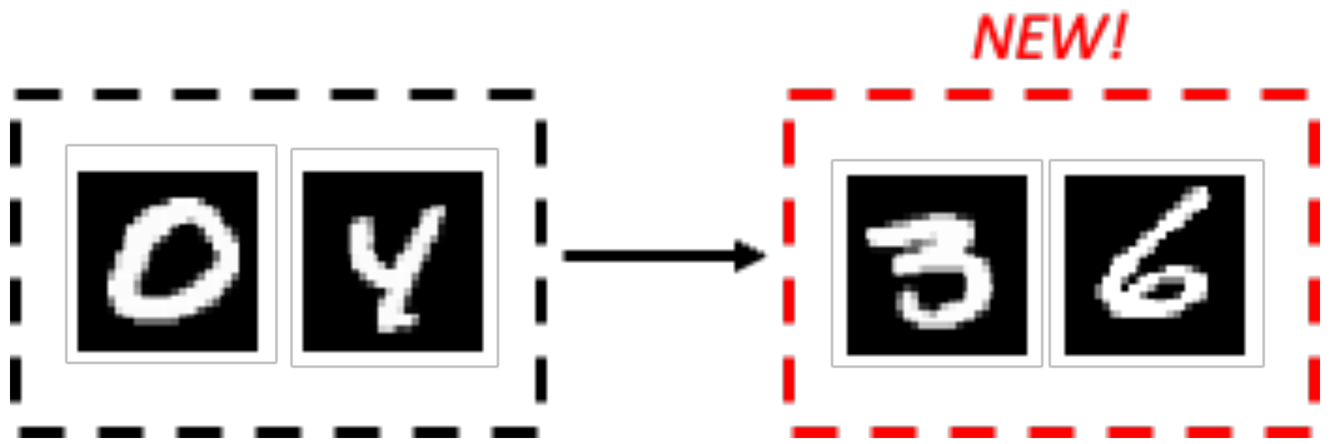
Forgets that which is **no longer needed**



Some Examples

Learn to classify **new classes**

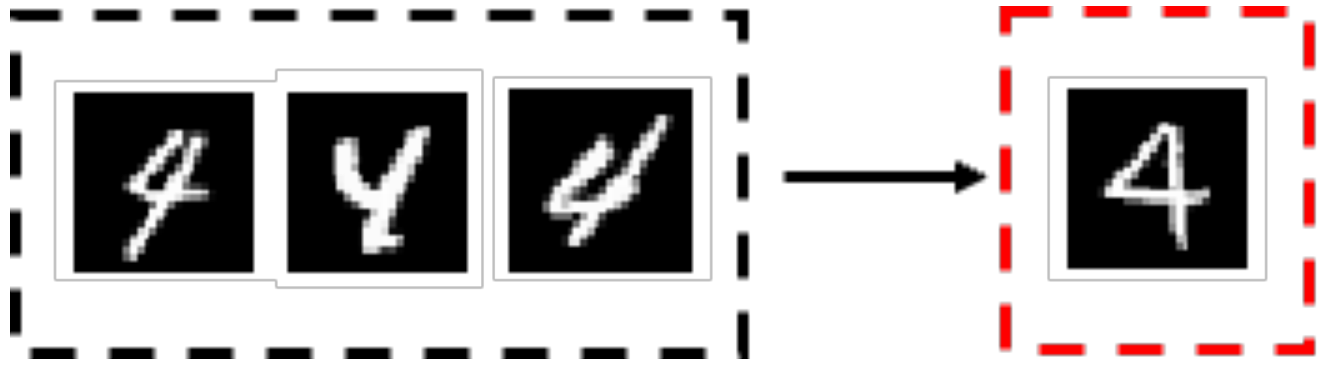
Related Research Areas / Jargon



Class Evolution (Stream Learning)

Class Incremental (Continual Learning)

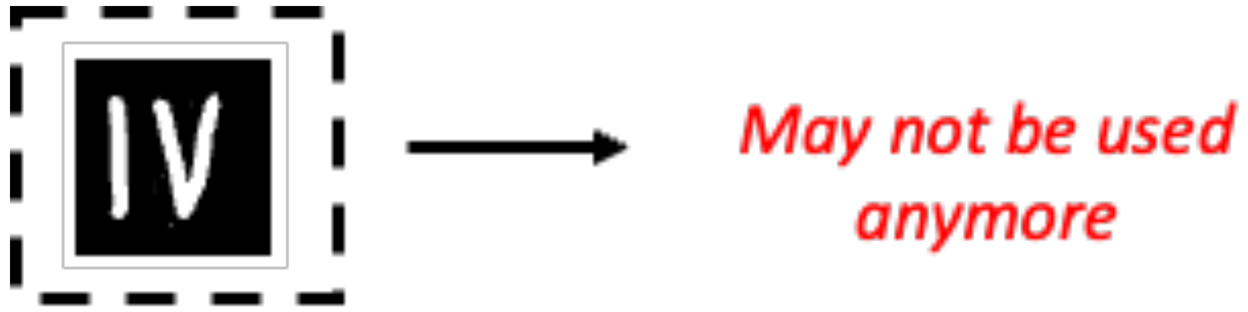
Update itself to accommodate for **changes within existing classes**



Concept Drift (Stream Learning)

Domain Incremental (Continual Learning)

Forgets that which is **no longer needed**



Class Evolution (Stream Learning)

The hidden context

The problem of concept drift: definitions and related work

Alexey Tsymbal
Department of Computer Science
Trinity College Dublin, Ireland
tsymbalo@tcd.ie

April 29, 2004

Abstract

In the real world concepts are often not stable but change with time. Typical examples of this are weather prediction rules and customers' preferences. The underlying data distribution may change as well. Often these changes make the model built on old data inconsistent with the new data, and regular updating of the model is necessary. This problem, known as *concept drift*, complicates the task of learning a model from data and requires special approaches, different from commonly used techniques, which treat arriving instances as equally important contributors to the final concept. This paper considers different types of concept drift, peculiarities of the problem, and gives a critical review of existing approaches to the problem.

"A difficult problem with learning in many real-world domains is that the concept of interest may depend on some **hidden context**, not given explicitly in the form of **predictive features.**"

TSYMBAL, 2004

Tsymbal, A., 2004. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin*

Widmer G., Kubat M., Learning in the presence of concept drift and hidden contexts, Machine Learning, 23 (1), 1996, 69-101.

Assumptions

Assumptions

Independent and identically distributed (iid)

Each data point in the stream **comes from the same probability distribution** **&**
The values of one data point **does not provide any information** about the values of another data point

Assumptions

Independent and identically distributed (iid)

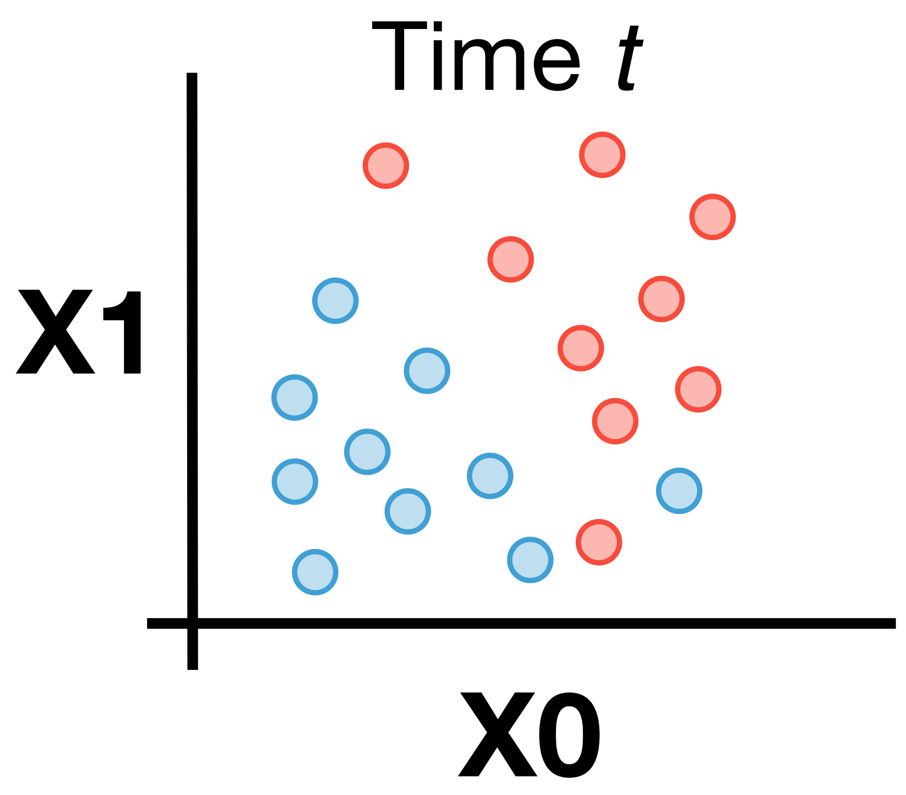
The presence of **Concept Drift (CD)** violates the **identically distributed** assumption

CD implies that **different sub-populations (concepts) exists** in the stream at different time intervals

Each concept **have its own statistical properties**

Concept Drift Example

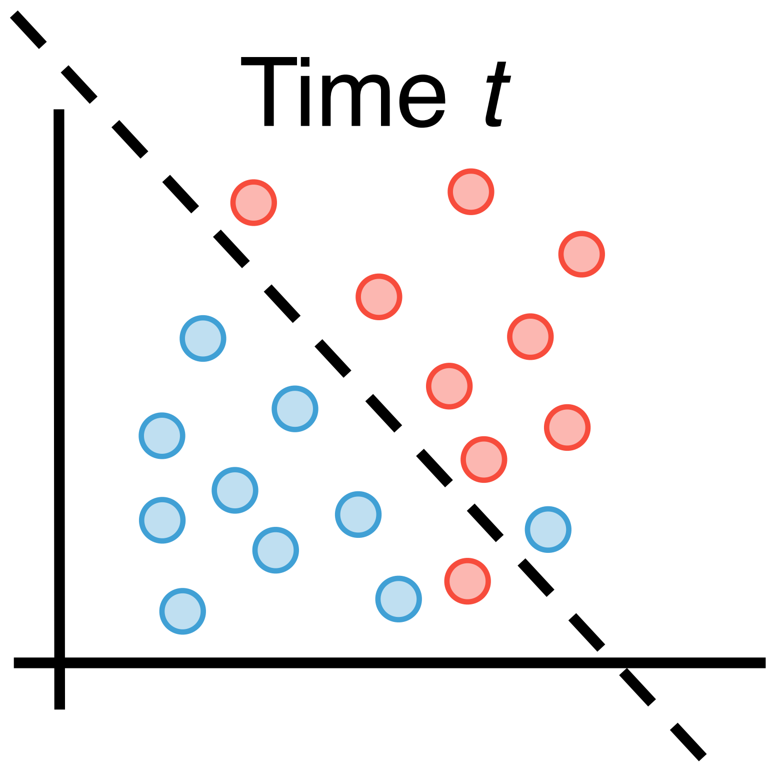
Concept drift example



Assume a simple classification problem

- Two classes 
- Two features (X0 and X1)

Concept drift

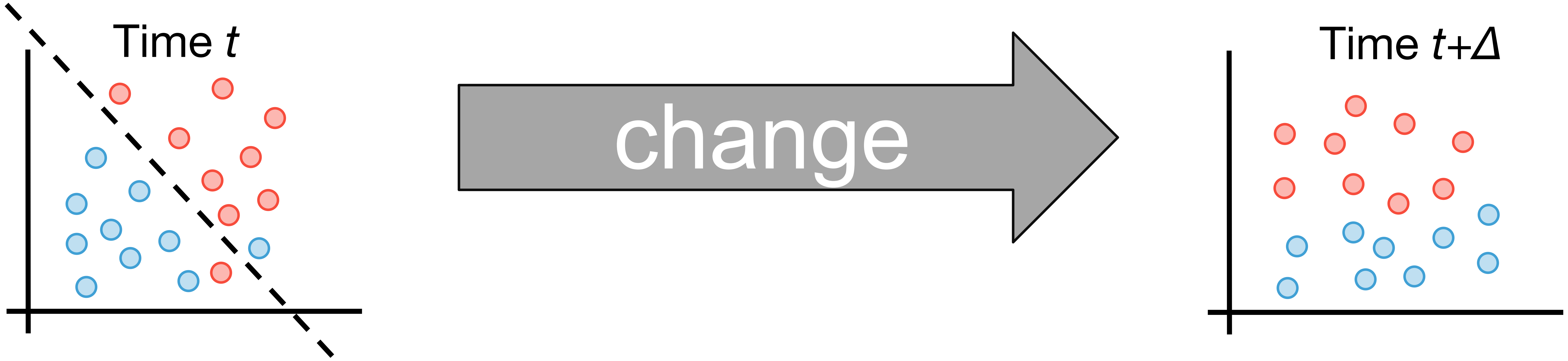


An accurate model!

We can build a very simple **linear model** to separate the two classes!

Concept drift

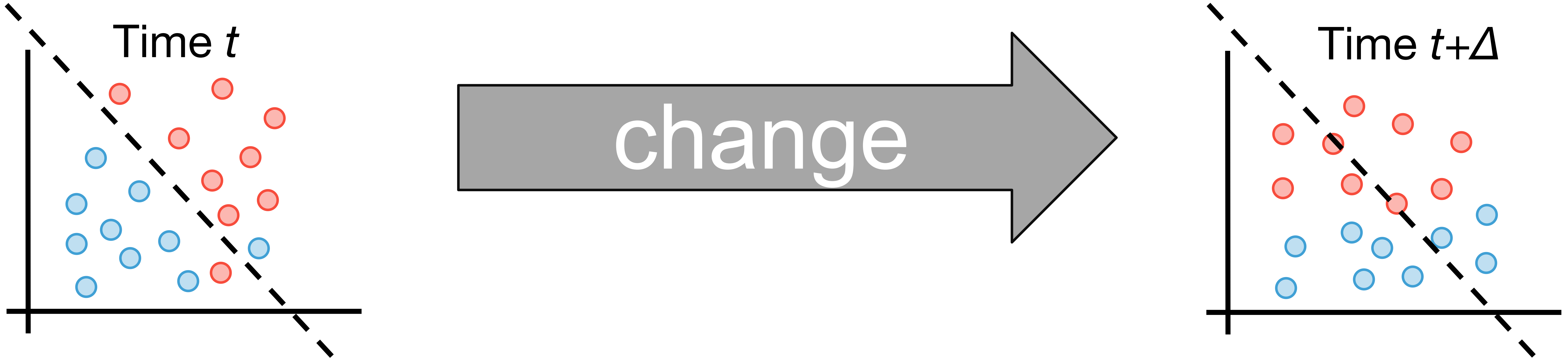
What if the data distribution **changes**?



An accurate model!

Concept drift

What if the data distribution **changes**?



An accurate model!

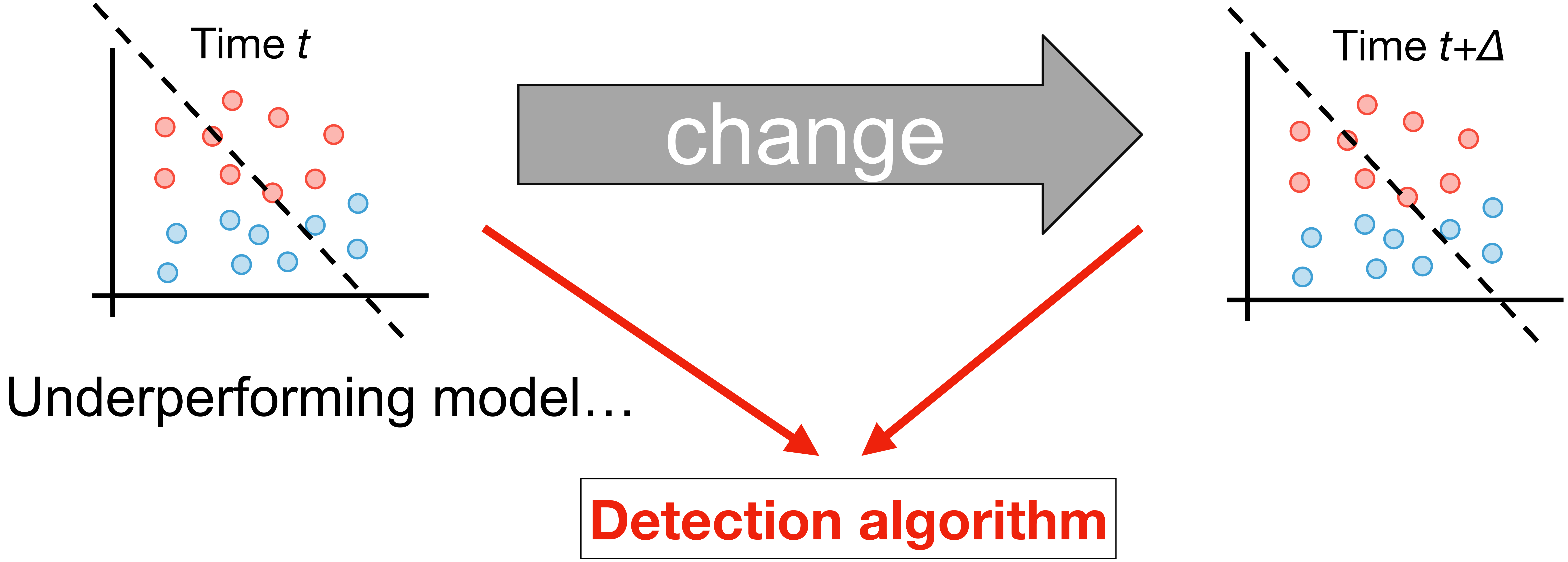
Not accurate anymore

What can we do about CD?

Detect & **Adapt** (update the model)

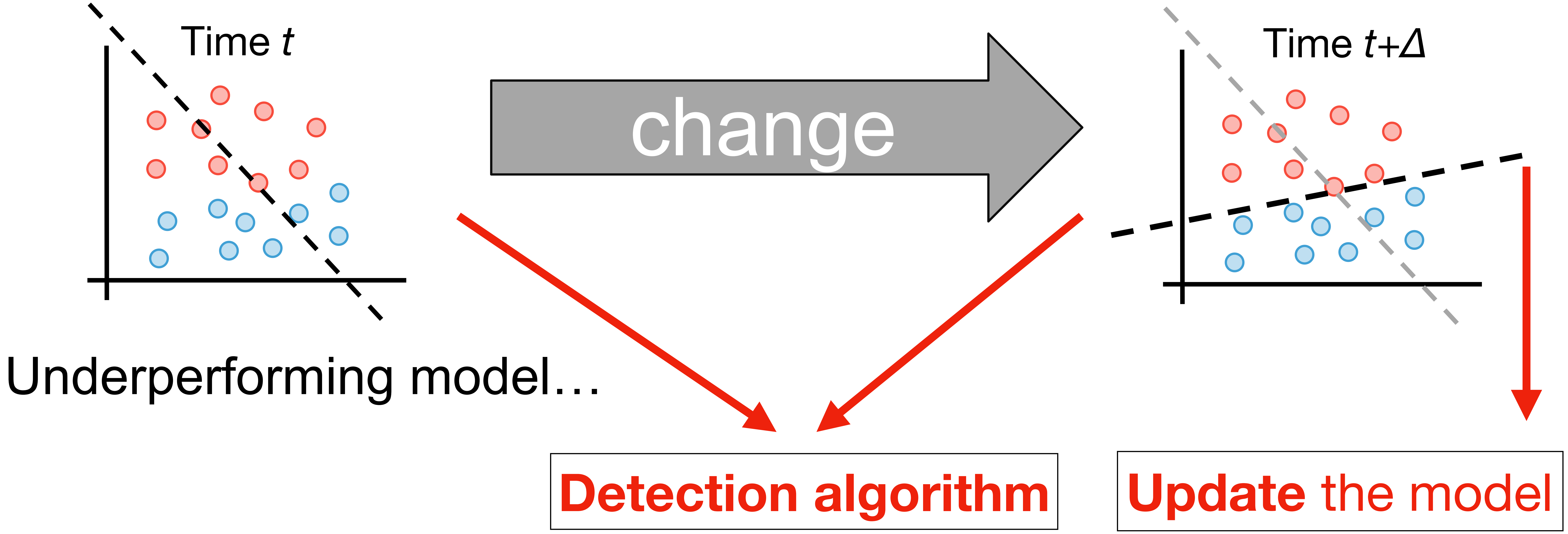
Concept drift

The data distribution may change overtime



Concept drift

The data distribution may change overtime

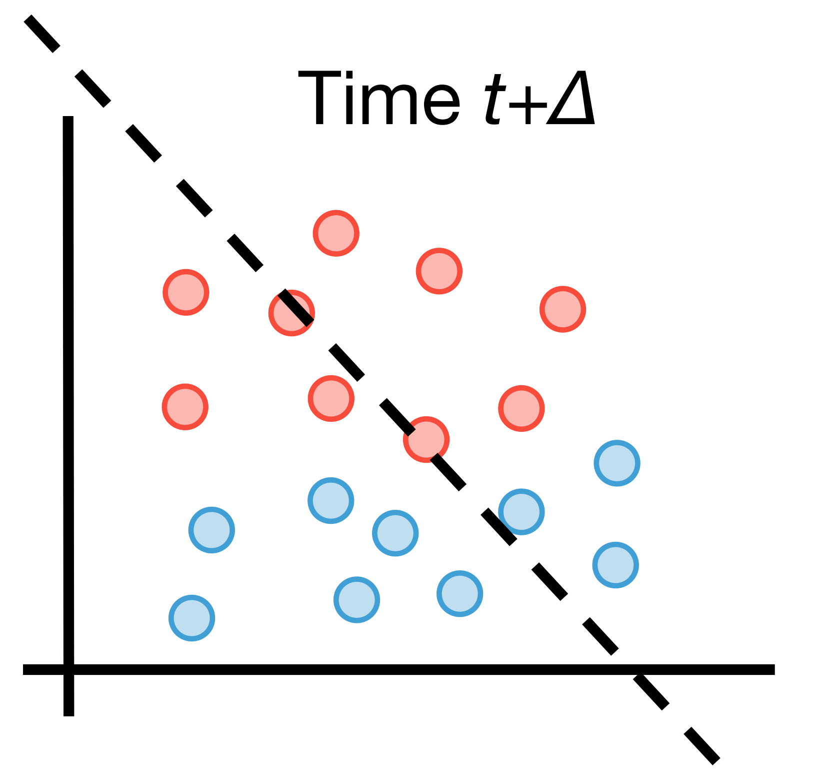


Concept drift

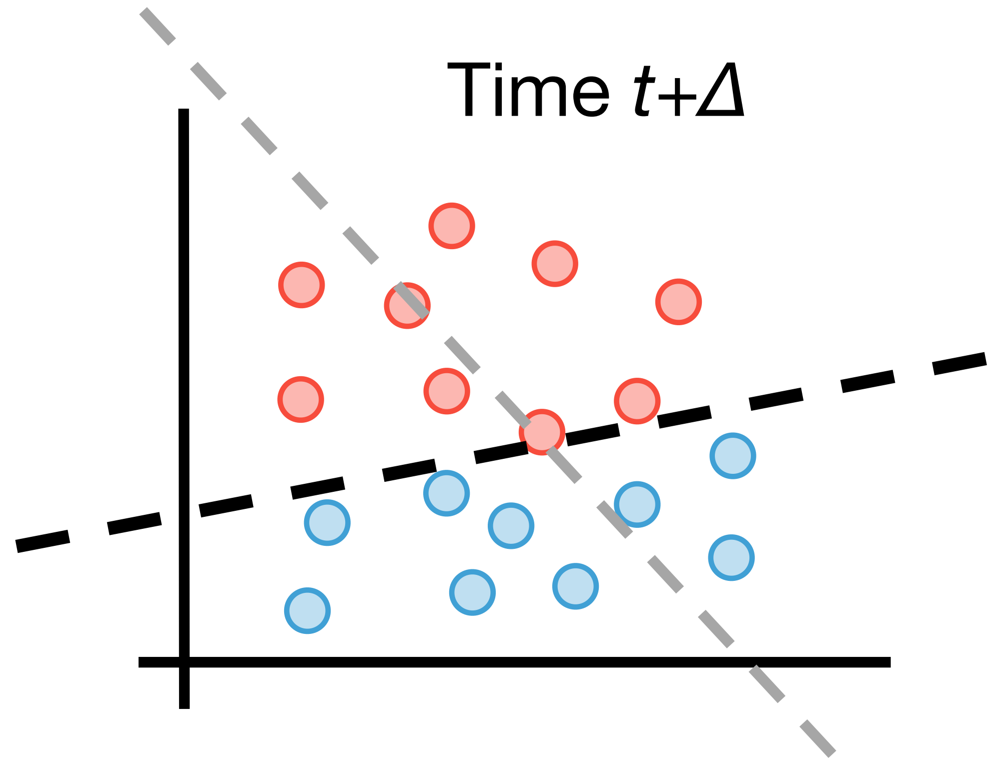
Some questions:

- What **data** should we use to train the updated model?

- How do we **detect** changes? What can the detection algorithm observe?



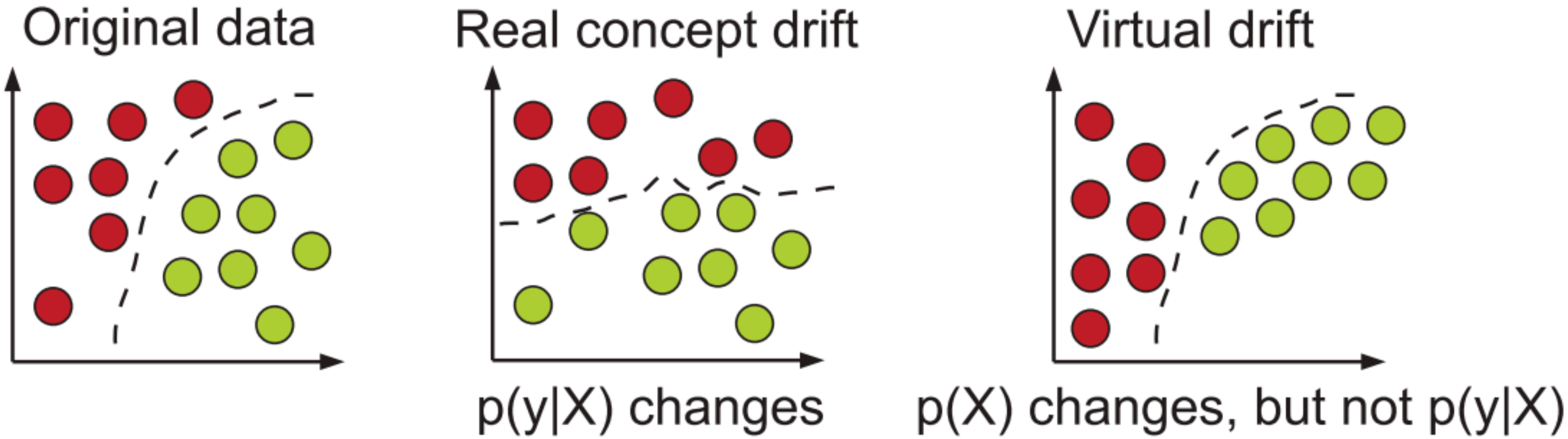
Underperforming model



Updated model

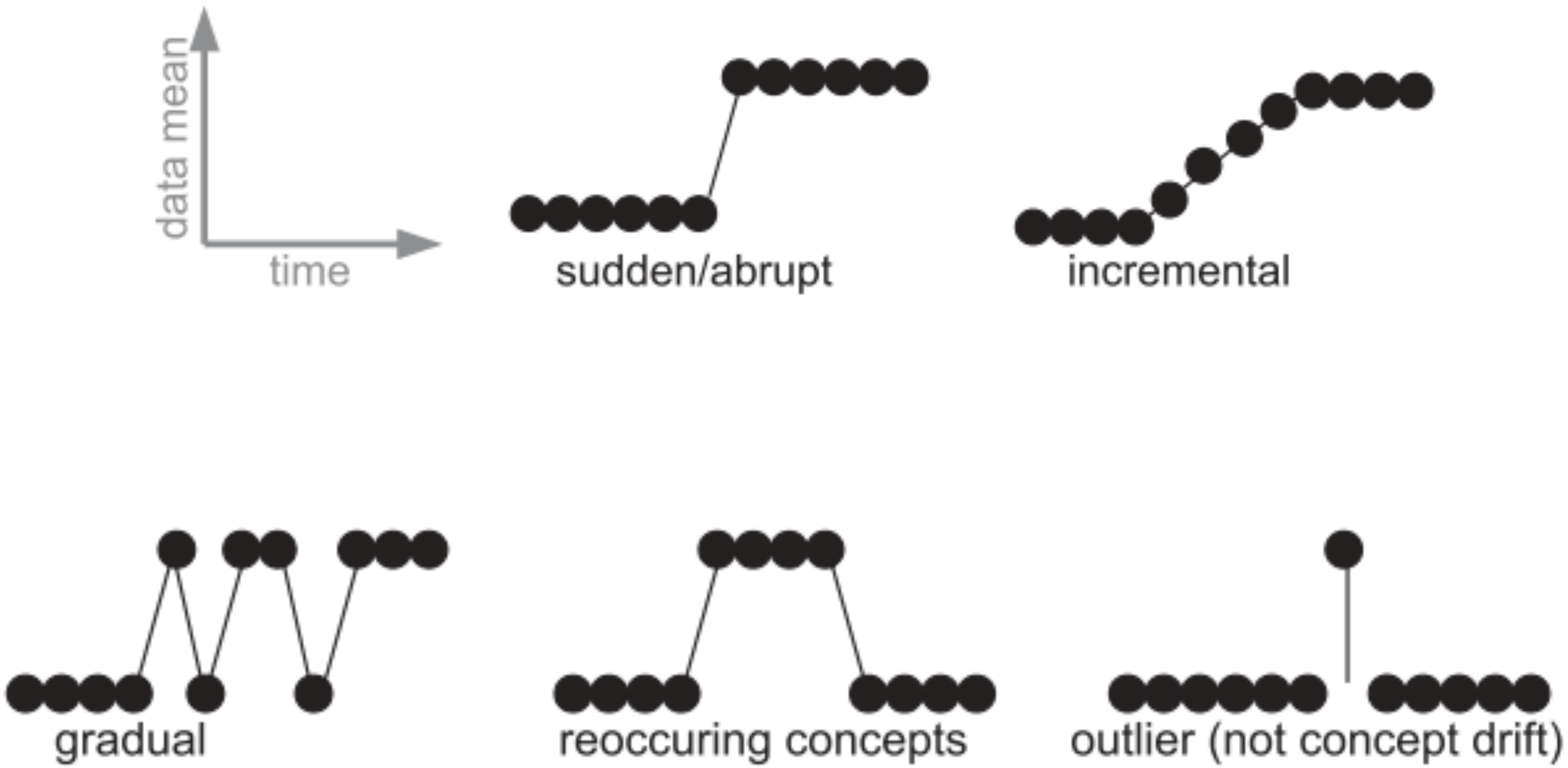
Categorising Concept Drift

Real x Virtual



Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*

Rate of change



Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*

ADWIN

ADaptive WINdow (ADWIN)

- Window based methods rely on a **window that sums up past data** and **a sliding window summarising recent data**

- Statistical tests are used to compare the distribution over the two windows

    - Null hypothesis: the distributions are equal

    - A rejection of the null hypothesis indicates a significant difference between the distributions of these windows (i.e. signals a change has happened)

Bifet, A., & Gavalda, R. (2007). Learning from time-changing data with adaptive windowing. *SIAM international conference on data mining*

ADaptive WINdow (ADWIN)

- Uses **sliding windows of variable size** that are recalculated online according to the rate of observed change of data in the windows

- Window is **increased** when there is no change, and **decreased** when a change has been detected

- ADWIN provides performance guarantees in the form of limits on <u>false positive rates</u> and <u>false negative rates</u>

- ADWIN doesn’t make assumptions about the underlying data distribution

Bifet, A., & Gavalda, R. (2007). Learning from time-changing data with adaptive windowing. *SIAM international conference on data mining*

Simulating CD

Why should we simulate?

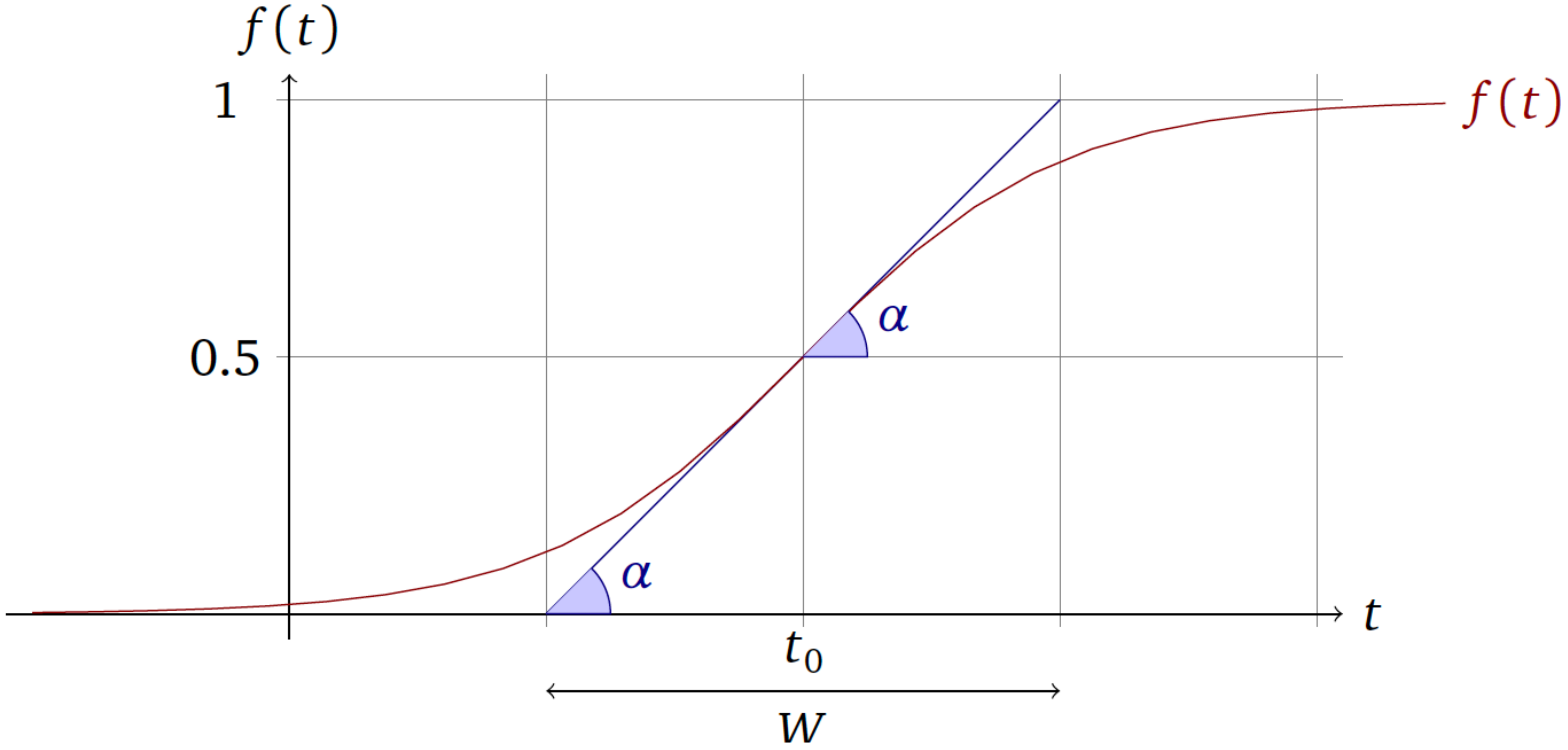
Concept drift is hard to define in a real data stream

Thus, studying it using real data can be challenging

One approach is to use synthetic data for studying and benchmarking algorithms

Concept Drift Framework

"Model a concept drift event as a **weighted combination of two pure distribution** that characterizes the target concepts before and after the drift." [Bifet et al, 2011]



A sigmoid function $f(t) = 1/(1 + e^{-s(t-t_0)})$.

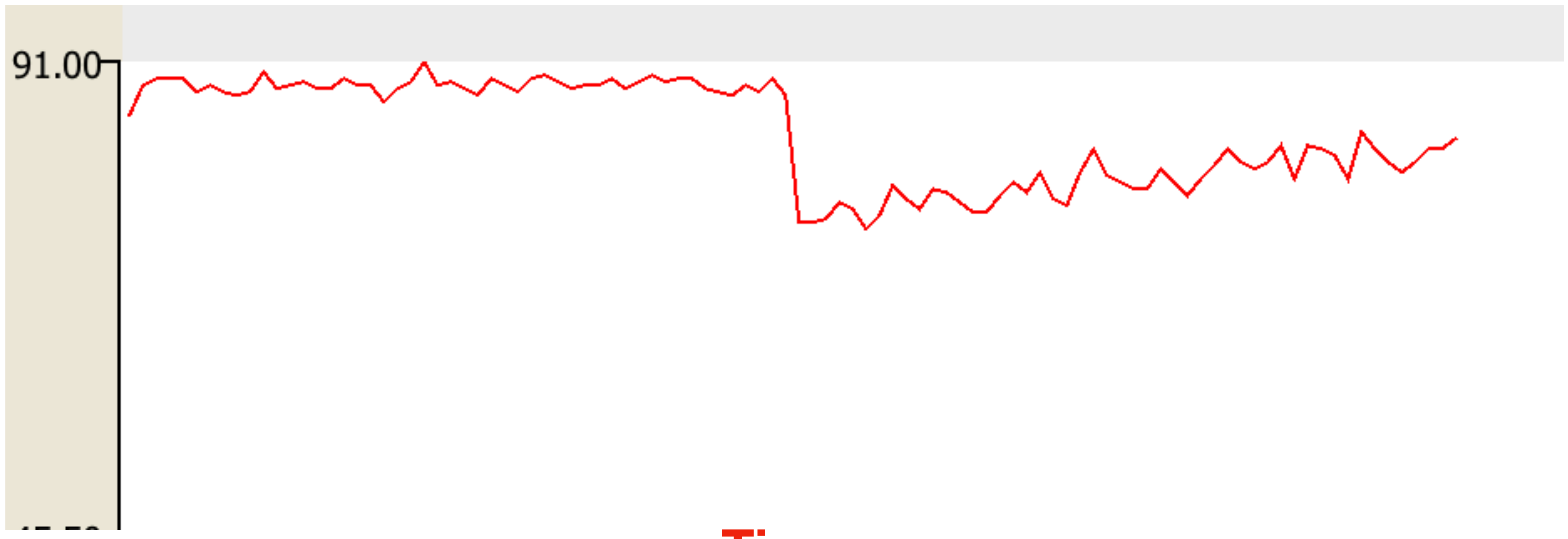
[Bifet et al, 2011] Bifet, A., & Kirkby, R. (2011). Data stream mining a practical approach. Chapter 2.7.1

Evaluation

Evaluating CD Detection

Common approach (proxy): **“Attach the method to a classifier, if the accuracy goes up, then the detector works”**

Accuracy



Time

Not necessarily the detector is successful in detecting changes, maybe it is just <u>randomly resetting the classifier</u>!

We must use specific metrics to evaluate a detector

Evaluating CD Detection

Important: we need the ground-truth of drift location for some of these

Some Metrics:

- Mean Time between False Alarms (MTFA)
- Mean Time to Detection (MTD)
- And others: MDR, ARL, MTR, …

Bifet, A. (2017). Classifier concept drift detection and the illusion of progress. In *Artificial Intelligence and Soft Computing ICAISC, 2017*

Hands-on example

KDD_2024_drift.ipynb