Handling Data Streams

Dr. Sonali Agarwal

Associate Professor

Big Data Analytics Lab

Indian Institute of Information Technology, Allahabad

Outline of the Lecture

- Handling Concept Drift
  - Background- Data Streams
  - Concept Drift
    - Definition and the Sources
    - Concept Drift Detection
    - Concept Drift Understanding
    - Drift Adaptation
    - Evaluation, Datasets, and Benchmarks
    - The Concept Drift Problem in other Research Areas
    - Conclusions: Findings and Future Directions

Background- Data Streams

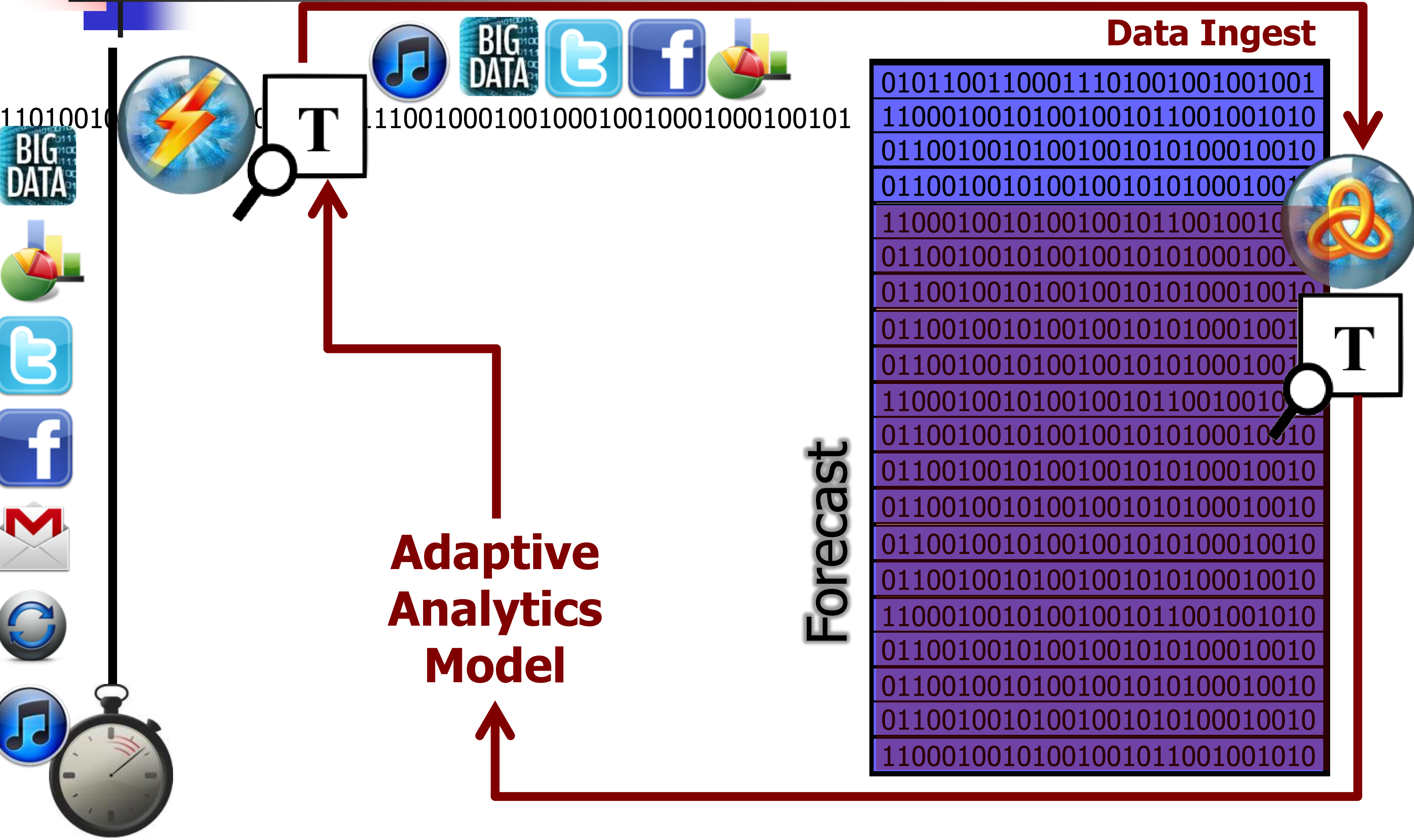
What is a stream?

- In traditional data processing applications, we know the entire dataset in advance
  - e.g. tables stored in a database
- Data streams are **high-volume, real-time** data that might be **unbounded**
  - we cannot store the entire stream in an accessible way
  - we have to process stream elements on-the-fly using limited memory

Properties of data streams

- They **arrive continuously** instead of being available a-priori
- They bear an arrival or a generation **timestamp**
- They are produced by external sources, i.e. the DSMS has **no control** over their **arrival order** or the **data rate**
- They have **unknown,** possibly **unbounded length,** i.e. the DSMS does not know when the stream ends

Analytic With *Data-In-Motion & Data At Rest*



Data Ingest

1101001

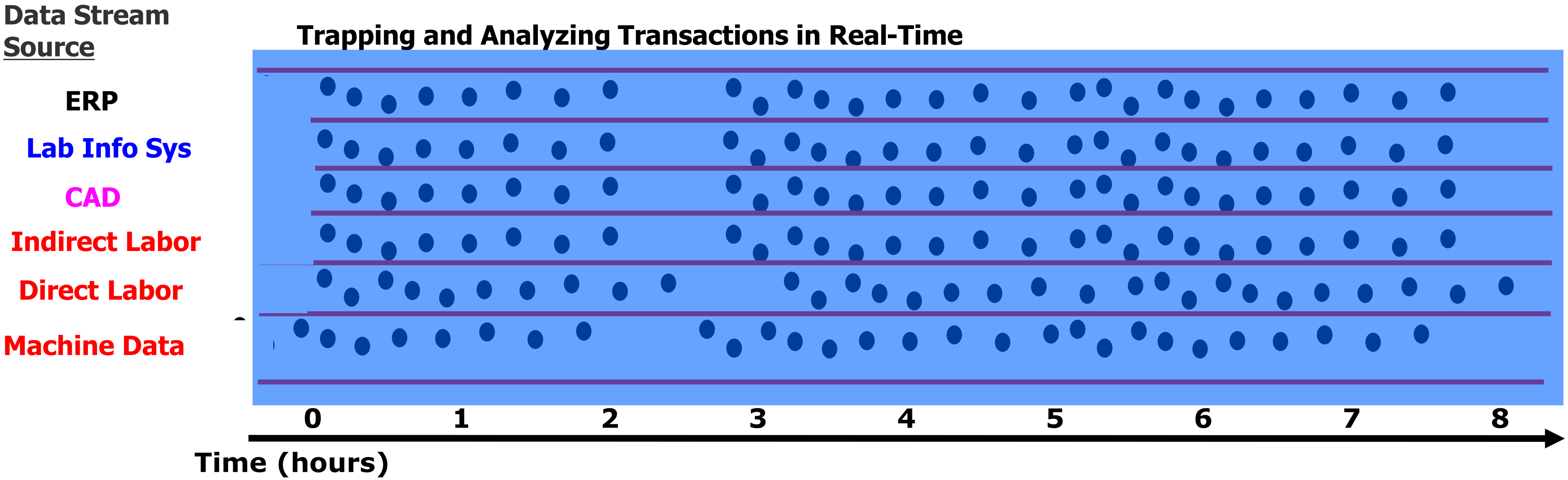
1110010001001000100100010001000100101

01011001100011101001001001001
11000100101001001011001001010
01100100101001001010100010010
0110010010100100101010001001
1100010010100100101100100101
0110010010100100101010001001
01100100101001001010100010010
0110010010100100101010001001
0110010010100100101010001001
1100010010100100101100100101
01100100101001001010100010010
01100100101001001010100010010
01100100101001001010100010010
01100100101001001010100010010
01100100101001001010100010010
11000100101001001011001001010
01100100101001001010100010010
01100100101001001010100010010
01100100101001001010100010010
11000100101001001011001001010

Forecast

Adaptive Analytics Model

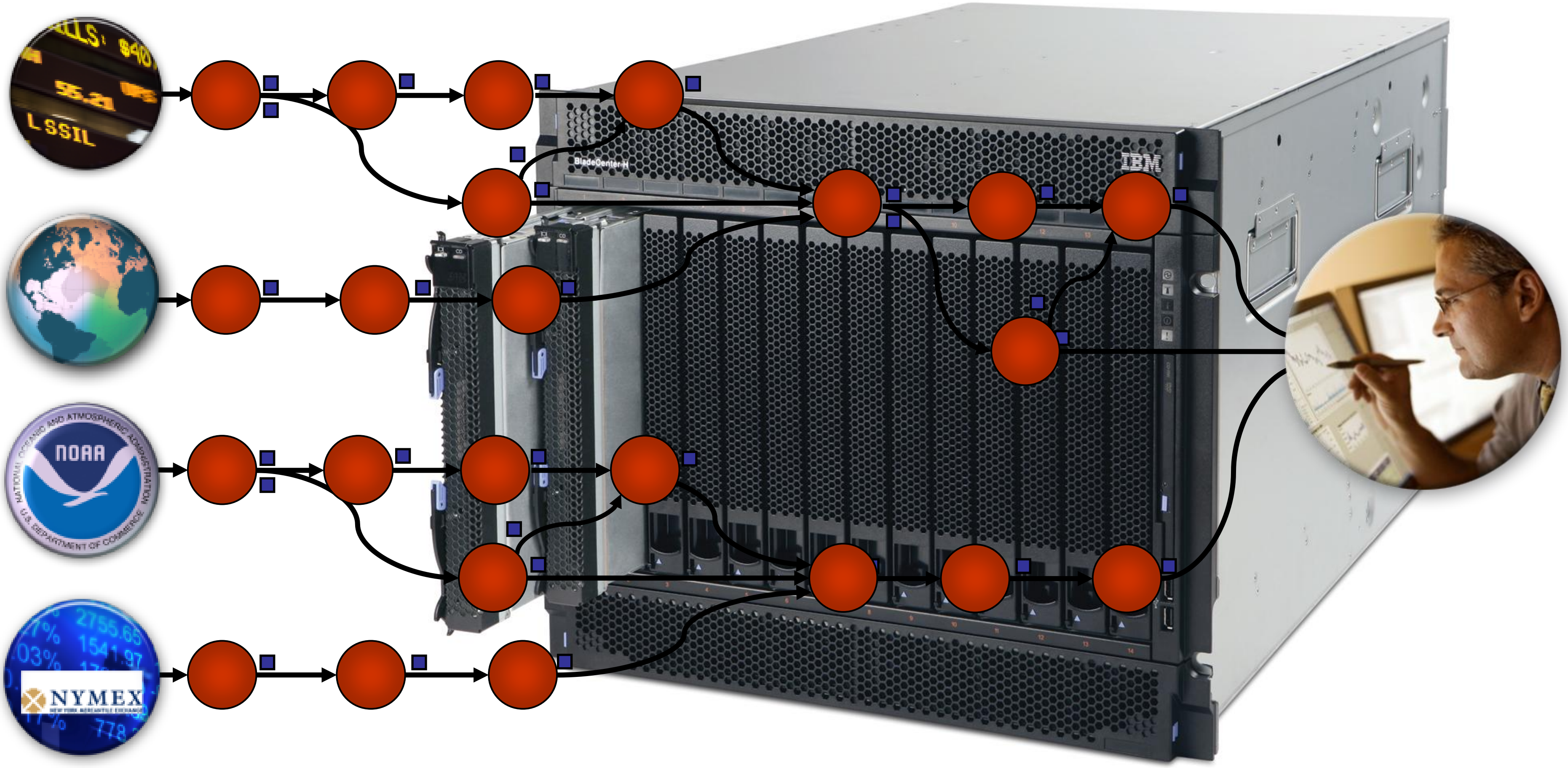
Real Time Data Stream



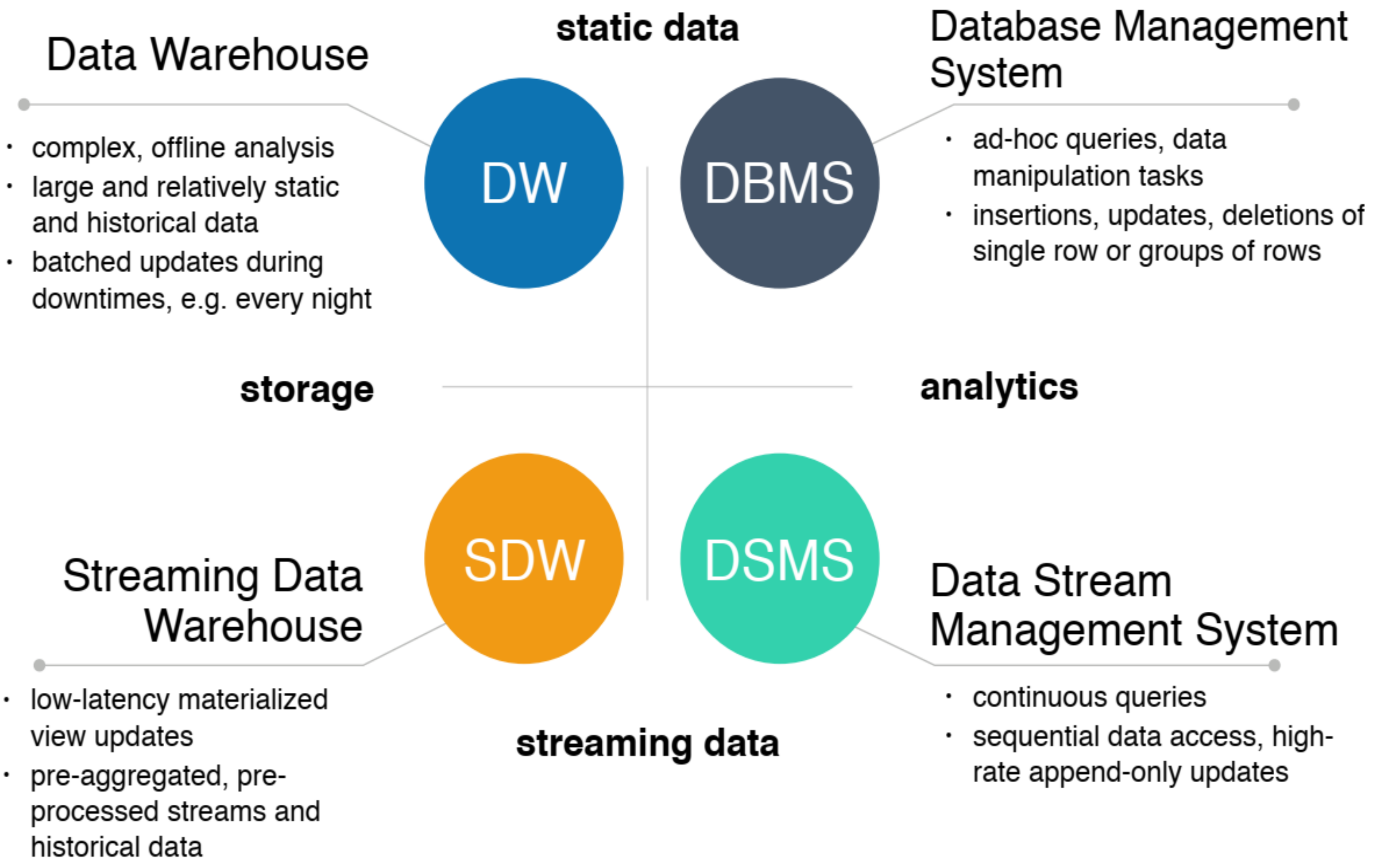
What is Stream Computing?

Continuous Ingestion

Continuous Analysis in Microseconds



Data Management Approaches



Data Warehouse

- complex, offline analysis
- large and relatively static and historical data
- batched updates during downtimes, e.g. every night

Database Management System

- ad-hoc queries, data manipulation tasks
- insertions, updates, deletions of single row or groups of rows

Streaming Data Warehouse

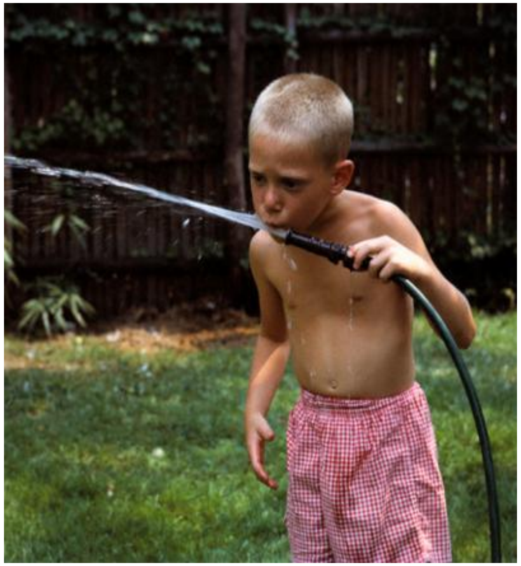
- low-latency materialized view updates
- pre-aggregated, pre-processed streams and historical data

Data Stream Management System

- continuous queries
- sequential data access, high-rate append-only updates

Batch Processing Vs Stream Processing

Batch Processing	Stream Processing
High-latency apps	Low-latency apps
Static Files	Event Streams
Process-after-store	Sense-and-respond
Batch processors	Stream processors



Example streams and applications

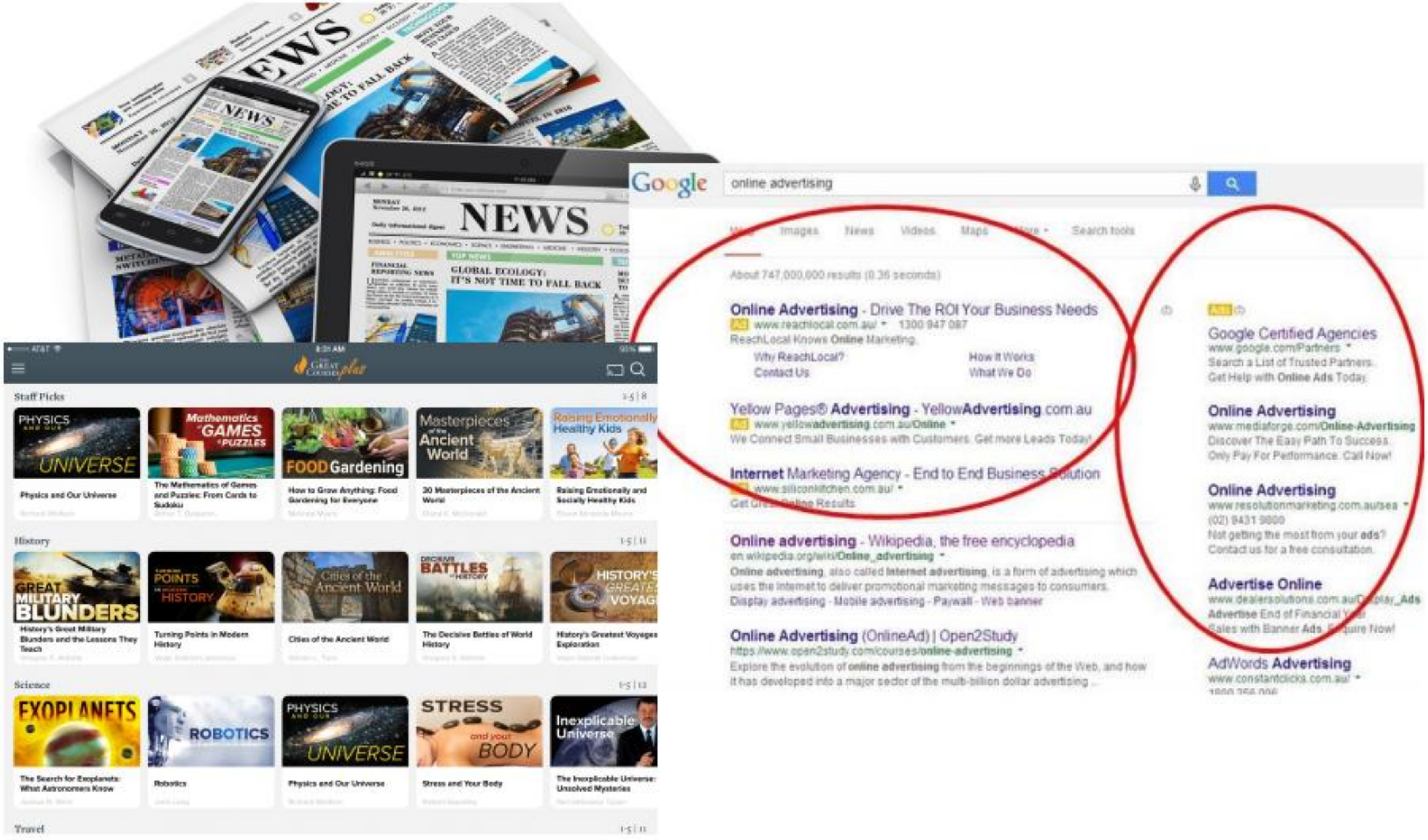
- Sensor measurements
  - anomaly detection, incident risk calculation

- Financial transactions
  - fraud detection, stock trading

- Vehicle location data and traffic data
  - report train system status, find optimal routes
- Web logs
  - online recommendations, personalization
- Network packets
  - intrusion detection, load balancing
- Online social interactions
  - trending topics, sentiment analysis

Location-based services

Online recommendations



AT&T 8:31 AM 93%

The Great Courses plus

Staff Picks 1-5 | 8

Physics and Our Universe

The Mathematics of Games and Puzzles: From Cards to Sudoku

How to Grow Anything: Food Gardening for Everyone

30 Masterpieces of the Ancient World

Raising Emotionally and Socially Healthy Kids

History 1-5 | 11

History's Great Military Blunders and the Lessons They Teach

Turning Points in Modern History

Cities of the Ancient World

The Decisive Battles of World History

History's Greatest Voyages Exploration

Science 1-5 | 12

The Search for Exoplanets: What Astronomers Know

Robotics

Physics and Our Universe

Stress and Your Body

The Inexplicable Universe: Unsolved Mysteries

Travel 1-5 | 11

Google online advertising

Web Images News Videos Maps More Search tools

About 747,000,000 results (0.36 seconds)

Online Advertising - Drive The ROI Your Business Needs
Ad www.reachlocal.com.au/ 1300 947 087
ReachLocal Knows **Online** Marketing.
Why ReachLocal? How It Works
Contact Us What We Do

Yellow Pages® **Advertising** - Yellow**Advertising**.com.au
Ad www.yellowadvertising.com.au/Online
We Connect Small Businesses with Customers. Get more Leads Today!

Internet Marketing Agency - End to End Business Solution
Ad www.siliconkitchen.com.au/
Get Great Online Results

Online advertising - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Online_advertising
Online advertising, also called **internet advertising**, is a form of advertising which uses the Internet to deliver promotional marketing messages to consumers.
Display advertising - Mobile advertising - Paywall - Web banner

Online Advertising (OnlineAd) | Open2Study
https://www.open2study.com/courses/online-advertising
Explore the evolution of **online advertising** from the beginnings of the Web, and how it has developed into a major sector of the multi-billion dollar advertising ...

Ads

Google Certified Agencies
www.google.com/Partners
Search a List of Trusted Partners.
Get Help with **Online Ads** Today.

Online Advertising
www.mediaforge.com/Online-Advertising
Discover The Easy Path To Success.
Only Pay For Performance. Call Now!

Online Advertising
www.resolutionmarketing.com.au/sea
(02) 8431 9000
Not getting the most from your **ads**?
Contact us for a free consultation.

Advertise Online
www.dealersolutions.com.au/Display_Ads
Advertise End of Financial Year
Sales with Banner **Ads**. Enquire Now!

AdWords **Advertising**
www.constantclicks.com.au/
1800 356 006

Sensor measurements analysis

- Monitoring applications
- Complex filtering and alarm activation
- Aggregation of multiple sensors and joins
- Examples
  - real-time statistics, e.g. weather maps
  - monitor conditions to adjust resources, e.g. power generation
  - energy monitoring for billing purposes

Stock trading

- Stock trading
  - discover correlations
  - identify trends
  - forecast future values
- Examples
  - Find all stocks priced between $20 and $200, where the spread between the high tick and the low tick over the past 30 minutes is greater than 3% of the last price, and where in the last 5 minutes the average volume has surged by more than 300%.
  - Find all stocks trading above their 200-day moving average with a market cap greater than $5 Billion that have gained in price today by at least 2%, and are within 2% of today’s high.

Financial transaction analysis

- Monitor transactions to detect fraud
  - online risk calculation, e.g. rules, clustering, regression, k-NN, neural networks
- Example: Someone steals your phone and sign in in your banking app. The app allows transfers of up to €1000 and so the thief makes transfers of €1000 to a "fake account" until either you're out of money or the activity is detected.
- Features to detect fraudulent activity like this:
  - The transaction amount.
  - The number of recent (e.g. the last hour) transactions.
  - Whether money was sent to this recipient account for the first time in the past 24 hours (in other words, to an “unknown” recipient account).

Call monitoring

- Service monitoring
  - source and destination phone numbers
  - their first and last cell towers
- Examples:
  - Location-based services
  - Monitor cell tower load
  - Continuously maintain call signatures for fraud detection
    - call frequency
    - top-K cell towers used

Web activity analysis

- Visualization
  - Parse and aggregate online logs: impressions, clicks, transactions, likes, comments
- • Analytics on user activity
  - Filtering, aggregation, joins with static data (e.g. user profile data)
- Examples
  - online A/B testing
  - trending topics
  - sentiment analysis, e.g., reaction to just published campaign
  - online recommendations of products, articles, people

Online traffic management

- Analysis of real-time vehicle locations to improve traffic conditions
- Provide real-time scheduling information for public transport
- Optimize transport network flow and recommend alternative routes

Example:

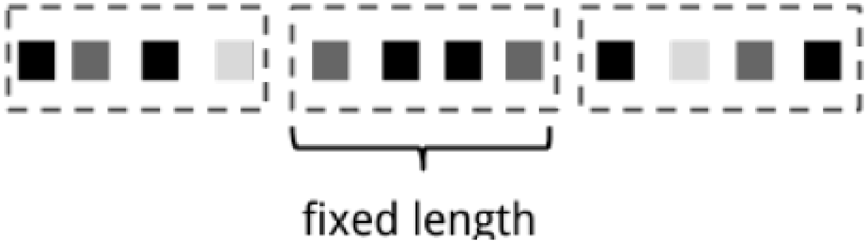
- Adjusts traffic lights in real-time to reduce congestion and clear paths for emergency response vehicles.

Requirements for streaming systems

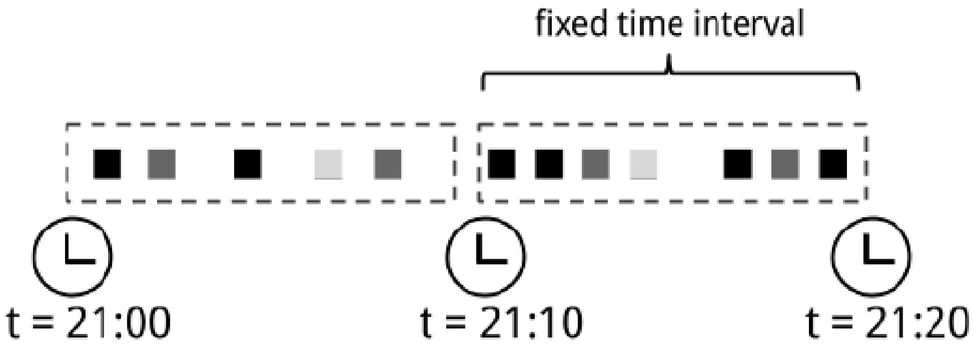
- Process events **online** without storing them
- Support a **high-level language** (e.g. StreamSQL)
- Handle missing, **out-of-order**, delayed data
- Guarantee **deterministic** (on replay) and **correct** results (on recovery)
- Combine **batch** (historical) and **stream processing**
- Ensure **availability** despite failures
- Support **distribution** and automatic **elasticity**
- Offer **low-latency**

Types of Windows

- **Tumbling** windows assign events into non-overlapping buckets of fixed size.



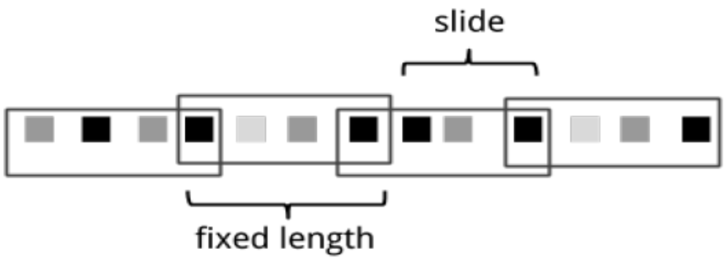
Count Based Window



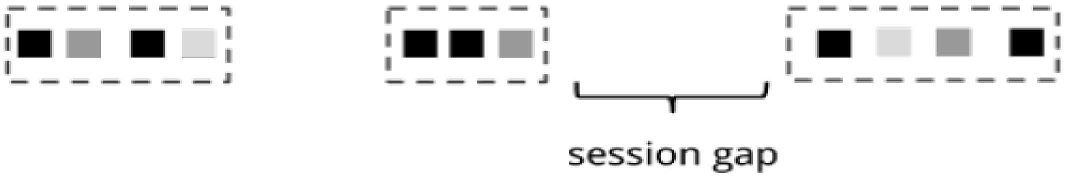
Time Based Window

Types of Windows

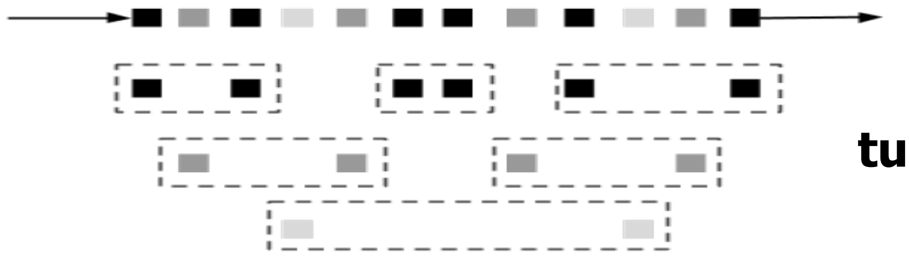
- **Sliding** windows assign events into overlapping buckets of fixed size.



Sliding Window



Session Window

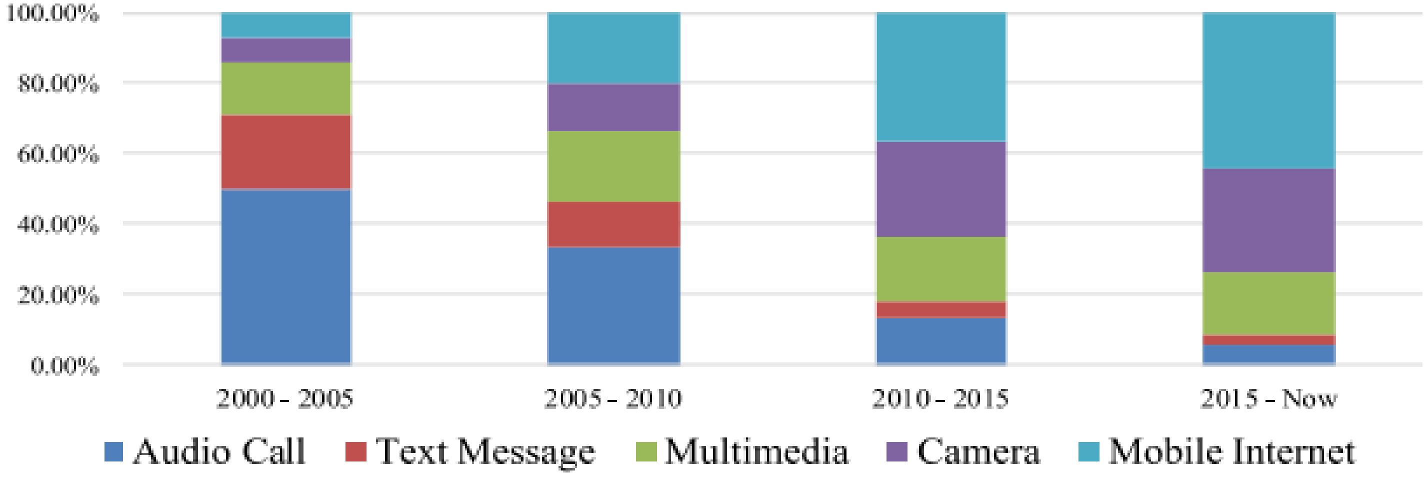


A parallel count-based tumbling window of length 2.

Concept Drift

Concept Drift Problem

- Concept drift problem exists in many real-world situations.
- An example can be seen in the changes of behavior in mobile phone usage.
- From the bars in this figure, the time percentage distribution of the mobile phone usage pattern has changed from "Audio Call" to "Camera" and then to "Mobile Internet" over the past two decades.



Concept drift in mobile phone usage
(data used in figure are for demonstration only)

Concept Drift Definition and the Sources

- Concept drift is a phenomenon in which the **statistical properties of a target domain change over time in an arbitrary way.**
- It was first proposed by J. C. Schlimmer and R. H. Granger who aimed to point out that **noise data may turn to non-noise information at different time.**
- **These changes might be caused by changes in hidden variables which cannot be measured directly.**

Concept Drift Definition

- **During the classification task, a learning model L attempts to predict the class label y_i(i = 1, . . . , c) of the incoming instance X. This prediction is based on estimating the distribution D which represents the joint probability P(X, y_i).**

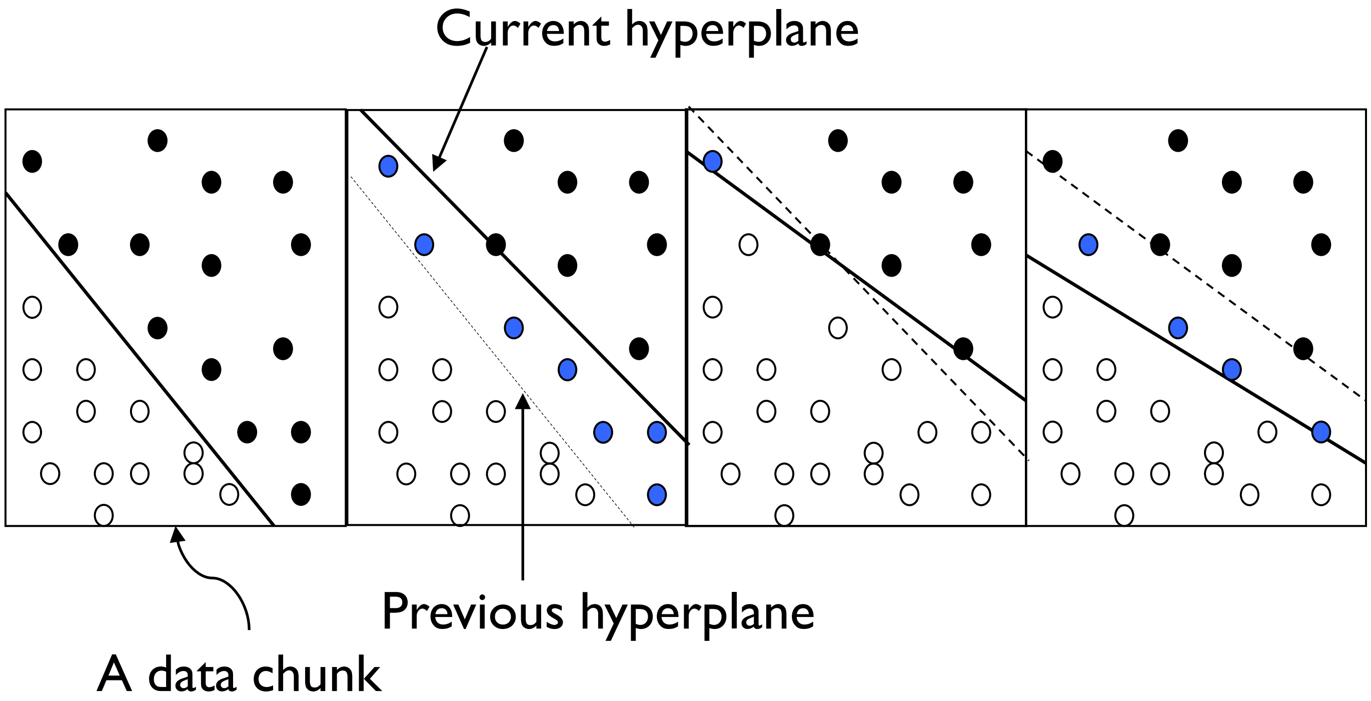
- Hence, when referring to a particular distribution D_t at time t (i.e. a particular joint probability P_t(X, y_i) at time t) we define it as a concept.

$$D_t = \{P_t(X, y_1), P_t(X, y_2), \ldots, P_t(X, y_c)\} \tag{1}$$

- **Thus, a concept drift occurs when there is a change in the joint probability between two time points t_0 and t_1:**

$$P_{t_0}(X, y_i) \neq P_{t_1}(X, y_i) \tag{2}$$

Concept-Drift



Current hyperplane

Previous hyperplane

A data chunk

Negative instance ●

Positive instance ○

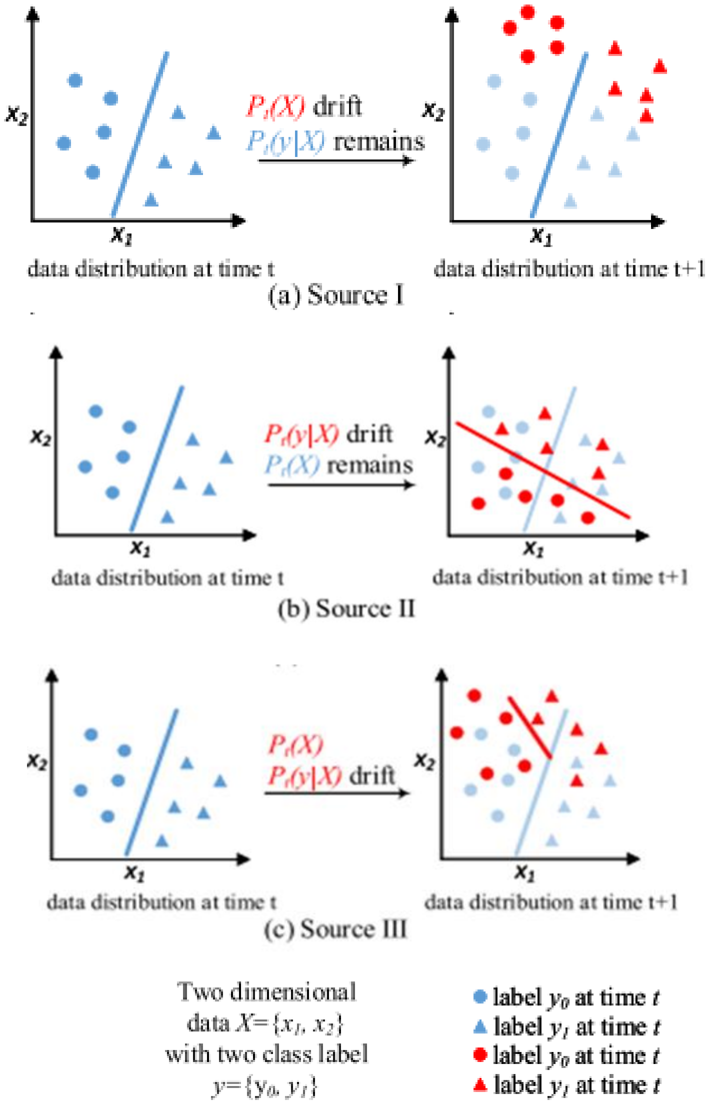
Instances victim of concept-drift ●

Three Sources of Concept Drift

Source I: $P_t(X) \neq P_{t+1}(X)$ while $P_t(y|X) = P_{t+1}(y|X)$, that is, the research focus is the drift in $P_t(X)$ while $P_t(y|X)$ remains unchanged. Since $P_t(X)$ drift does not affect the decision boundary, it has also been considered as virtual drift

Source II: $P_t(y|X) \neq P_{t+1}(y|X)$ while $P_t(X) = P_{t+1}(X)$ while $P_t(X)$ remains unchanged. This drift will cause decision boundary change and lead to learning accuracy decreasing, which is also called actual drift,

Source III: mixture of Source I and Source II, namely $P_t(X) \neq P_{t+1}(X)$ and $P_t(y|X) \neq P_{t+1}(y|X)$. Concept drift focus on the drift of both $P_t(y|X)$ and $P_t(X)$, since both changes convey important information about learning environment


x_2
x_1
data distribution at time t
$P_t(X)$ drift
$P_t(y|X)$ remains
data distribution at time t+1
$P_t(y|X)$ drift
$P_t(X)$ remains
$P_t(X)$
$P_t(y|X)$ drift
Two dimensional data $X=\{x_1, x_2\}$ with two class label $y=\{y_0, y_1\}$
● label y_0 at time t
▲ label y_1 at time t
● label y_0 at time t
▲ label y_1 at time t

(a) Source I

(b) Source II

(c) Source III

Example of Concept Drift

- **Real Concept Drift**
  - Let consider the Harry Potter film series where watchers have to consider them as interesting or junk.
  - Suppose that the watchers are adults who initially enjoyed the films for their special effects; after a long period of time, they may no longer enjoy them as their special effects become outdated.
  - This is known by **Real concept drift** because a change has occurred in the watcher preference.

Example of Concept Drift

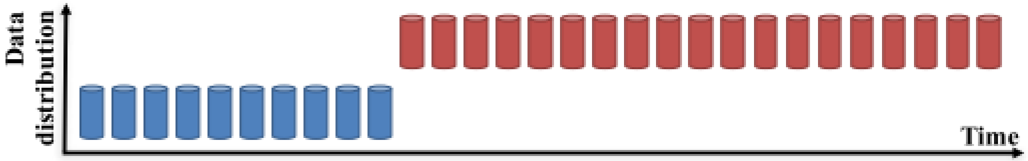
- **Virtual Concept Drift**
  - Another situation is when considering the watchers as children who are enjoying the films today; after a certain period of time, children grow up and their personal preferences grow up too.
  - Thus, if they are still enjoying the Harry Potter films, then we will consider this evolution as **Virtual concept drift** because it does not affect their preference.

Example of Concept Drift

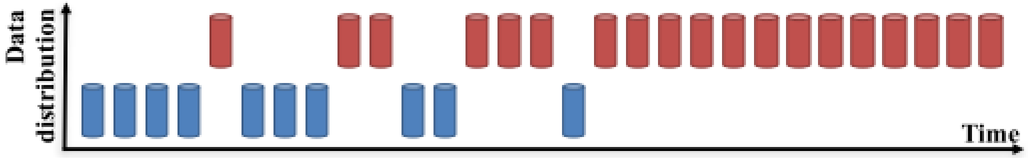
- **Class prior concept drift**
  - Finally, when the watchers may no longer be interested by the Harry Potter films because of the emergence of new wizard film series, this can considered as **Class prior concept drift.**

An example of concept drift types

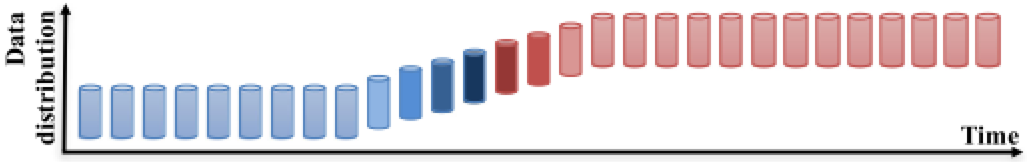
- Sudden Drift: A new concept occur within a short time.



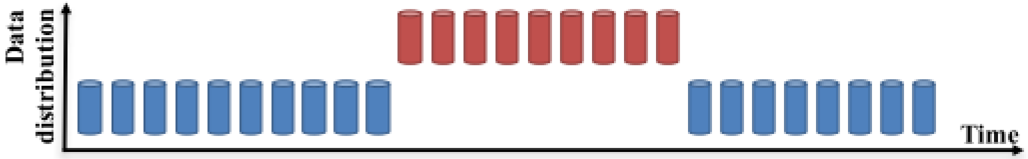
- Gradual Drift: A new concept gradually replaces the old one with respect to time.



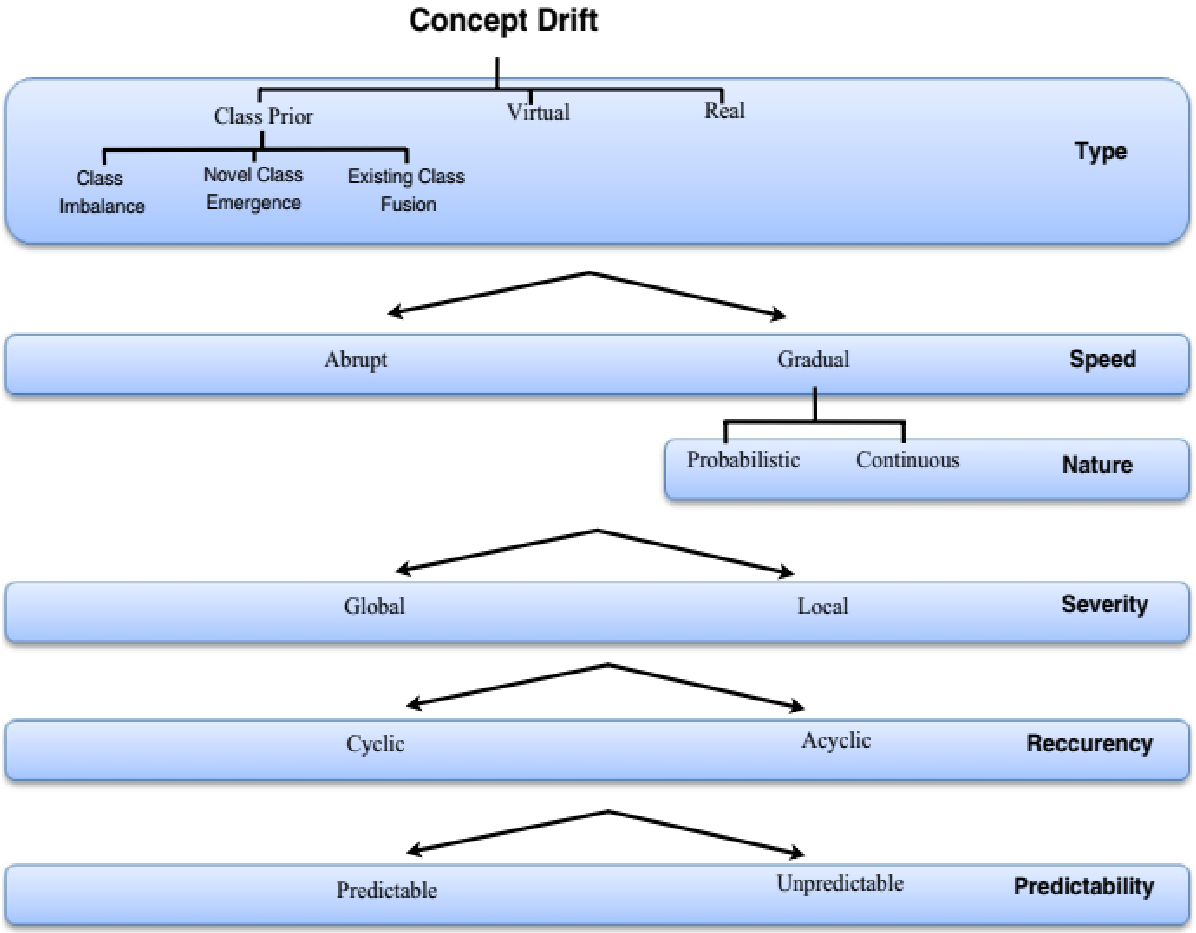
- Incremental Drift: An old concept incrementally changes to a new concept over a period of time.



- Reoccurring Concepts: An old concept reoccur after sometime



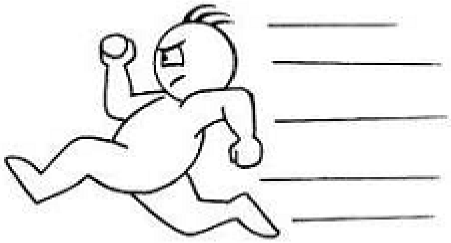
Concept drift characteristics



- 1. How long does the drift last?
- 2. How does the new concept replace the old one?
- 3. How much change does the new concept cause?
- 4. Is the drift recurrent?
- 5. Is the drift predictable?

Desired Properties of a System to Handle Concept Drift

- Adapt to concept drift asap



- Distinguish noise from changes

Desired Properties of a System to Handle Concept Drift

- Robust to noise, but adaptive to changes
- Recognizing and reacting to reoccurring contexts

- Adapting with limited resources
  – time and memory

Criteria for evaluation of the ability of the algorithm to handle concept drift:

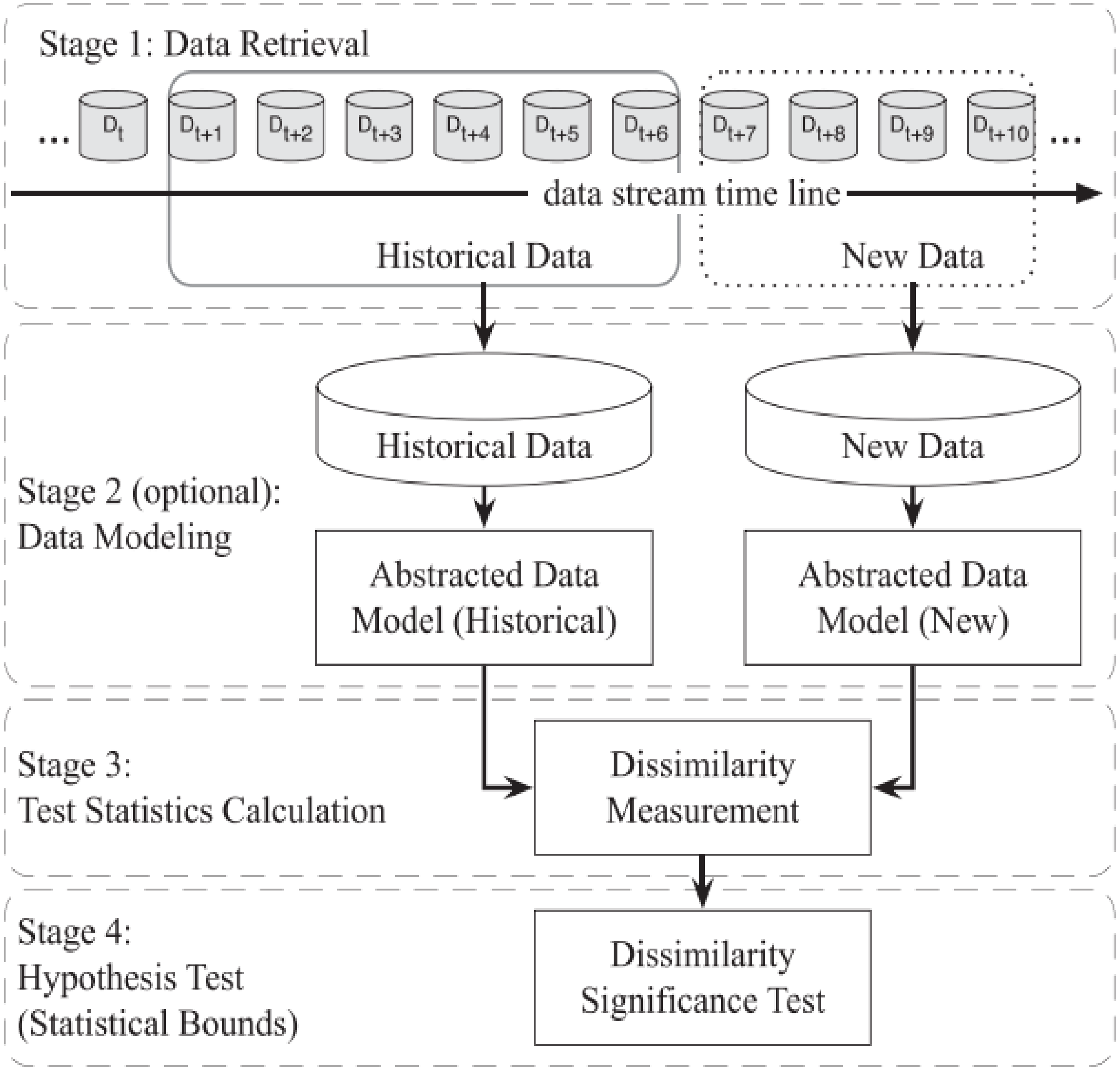
Learning in Presence of Concept Drift

- Criteria for evaluation of the ability of the algorithm to handle concept drift:
  - Delay **[Reflects how fast the method can detect/adapt to the concept drift.]**
  - Resistance to noise. **[Characterizes the ability of the method to distinguish the noise in the data from the real concept drift.]**
  - Cost of adaptation. **[Defines whether the algorithm needs to recompute the model from scratch after detecting the concept drift, or the localized re-computation is enough]**

Concept Drift Detection

Drift detection refers to the techniques and mechanisms that characterize and quantify concept drift via identifying change points or change time intervals.

A general framework for drift detection contains four stages, as shown in Figure

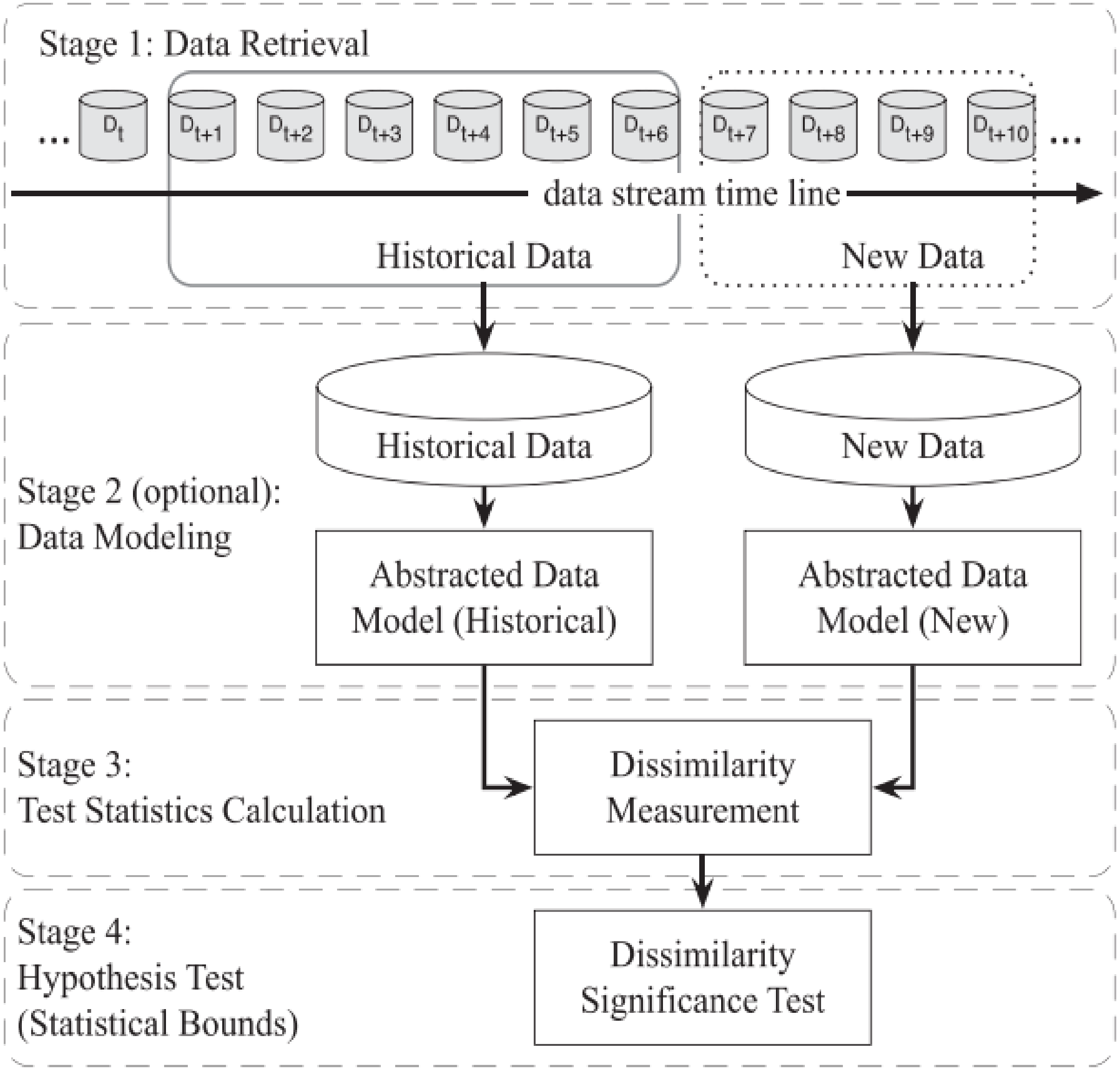


Stage 1: Data Retrieval
...
D_t
D_{t+1}
D_{t+2}
D_{t+3}
D_{t+4}
D_{t+5}
D_{t+6}
D_{t+7}
D_{t+8}
D_{t+9}
D_{t+10}
...
data stream time line
Historical Data
New Data
Stage 2 (optional): Data Modeling
Historical Data
New Data
Abstracted Data Model (Historical)
Abstracted Data Model (New)
Stage 3: Test Statistics Calculation
Dissimilarity Measurement
Stage 4: Hypothesis Test (Statistical Bounds)
Dissimilarity Significance Test

Concept Drift Detection

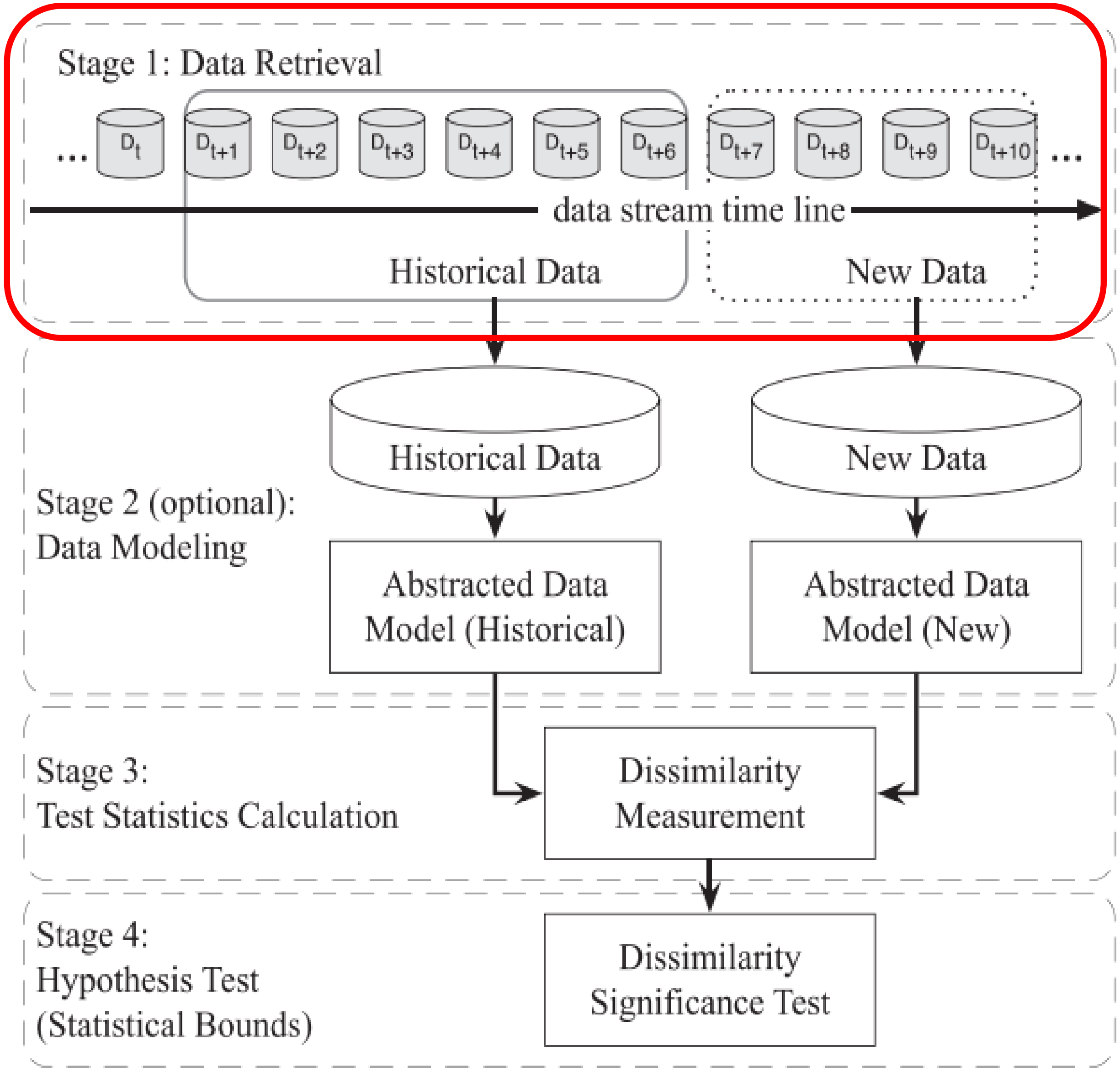
Drift detection refers to the techniques and mechanisms that characterize and quantify concept drift via identifying change points or change time intervals.

A general framework for drift detection contains four stages, as shown in Figure


Stage 1: Data Retrieval
...
D_t
D_{t+1}
D_{t+2}
D_{t+3}
D_{t+4}
D_{t+5}
D_{t+6}
D_{t+7}
D_{t+8}
D_{t+9}
D_{t+10}
...
data stream time line
Historical Data
New Data
Stage 2 (optional): Data Modeling
Historical Data
New Data
Abstracted Data Model (Historical)
Abstracted Data Model (New)
Stage 3: Test Statistics Calculation
Dissimilarity Measurement
Stage 4: Hypothesis Test (Statistical Bounds)
Dissimilarity Significance Test

Concept Drift Detection

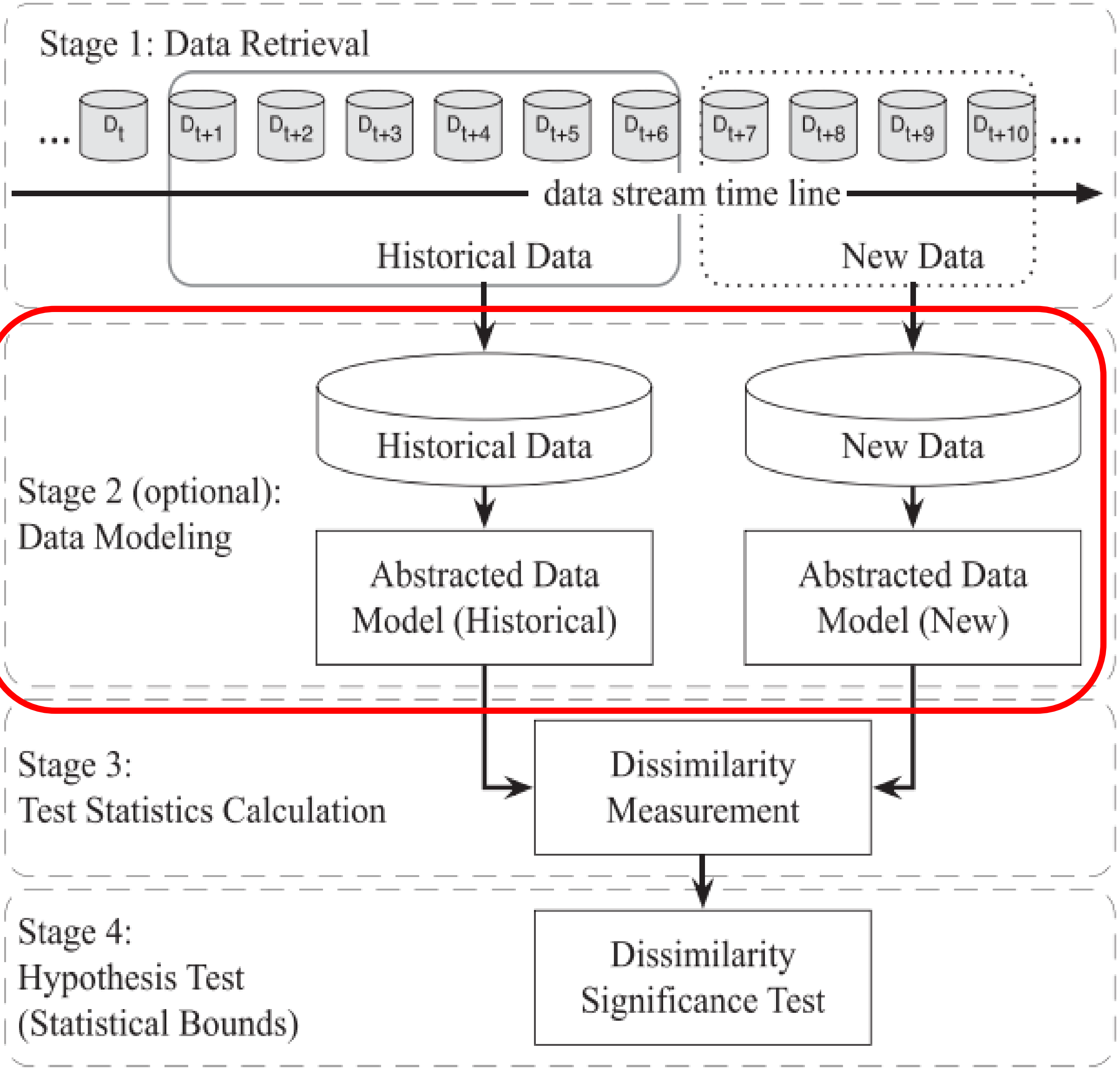
- Stage 1 (Data Retrieval) aims to retrieve data chunks from data streams.
- Since a single data instance cannot carry enough information to infer the overall distribution [2], knowing how to organize data chunks to form a meaningful pattern or knowledge is important in data stream analysis tasks.



Concept Drift Detection

- Stage 2 (Data Modeling) aims to abstract the retrieved data and extract the key features containing sensitive information, that is, the features of the data that most impact a system if they drift.

- This stage is optional, because it mainly concerns dimensionality reduction, or sample size reduction, to meet storage and online speed requirements.

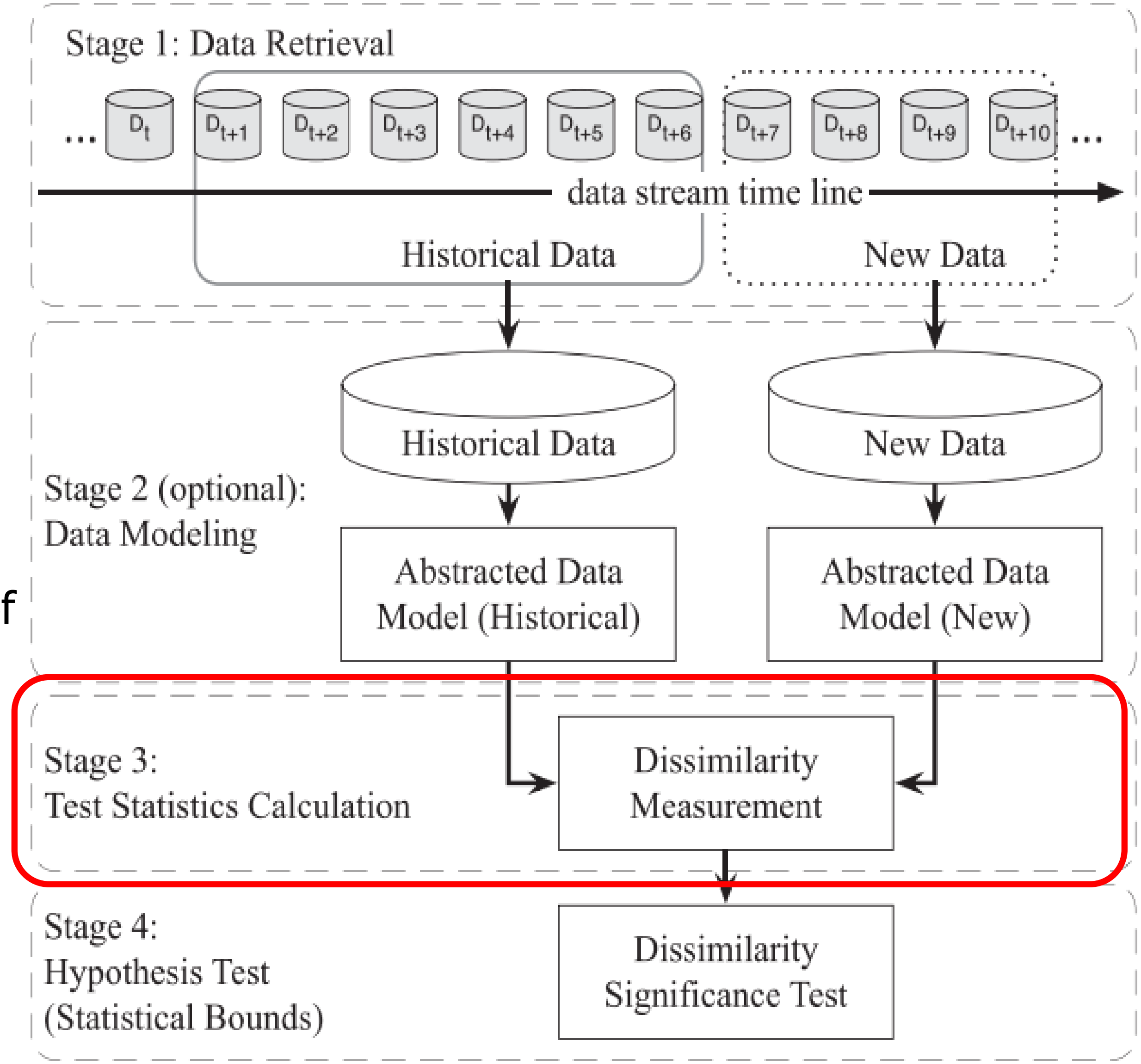

Stage 1: Data Retrieval
…
D_t
D_{t+1}
D_{t+2}
D_{t+3}
D_{t+4}
D_{t+5}
D_{t+6}
D_{t+7}
D_{t+8}
D_{t+9}
D_{t+10}
…
data stream time line
Historical Data
New Data
Historical Data
New Data
Stage 2 (optional): Data Modeling
Abstracted Data Model (Historical)
Abstracted Data Model (New)
Stage 3: Test Statistics Calculation
Dissimilarity Measurement
Stage 4: Hypothesis Test (Statistical Bounds)
Dissimilarity Significance Test

Concept Drift Detection

- Stage 3 (Test Statistics Calculation) is measurement of dissimilarity, or distance estimation. It quantifies the severity of the drift and forms test statistics for the hypothesis test.

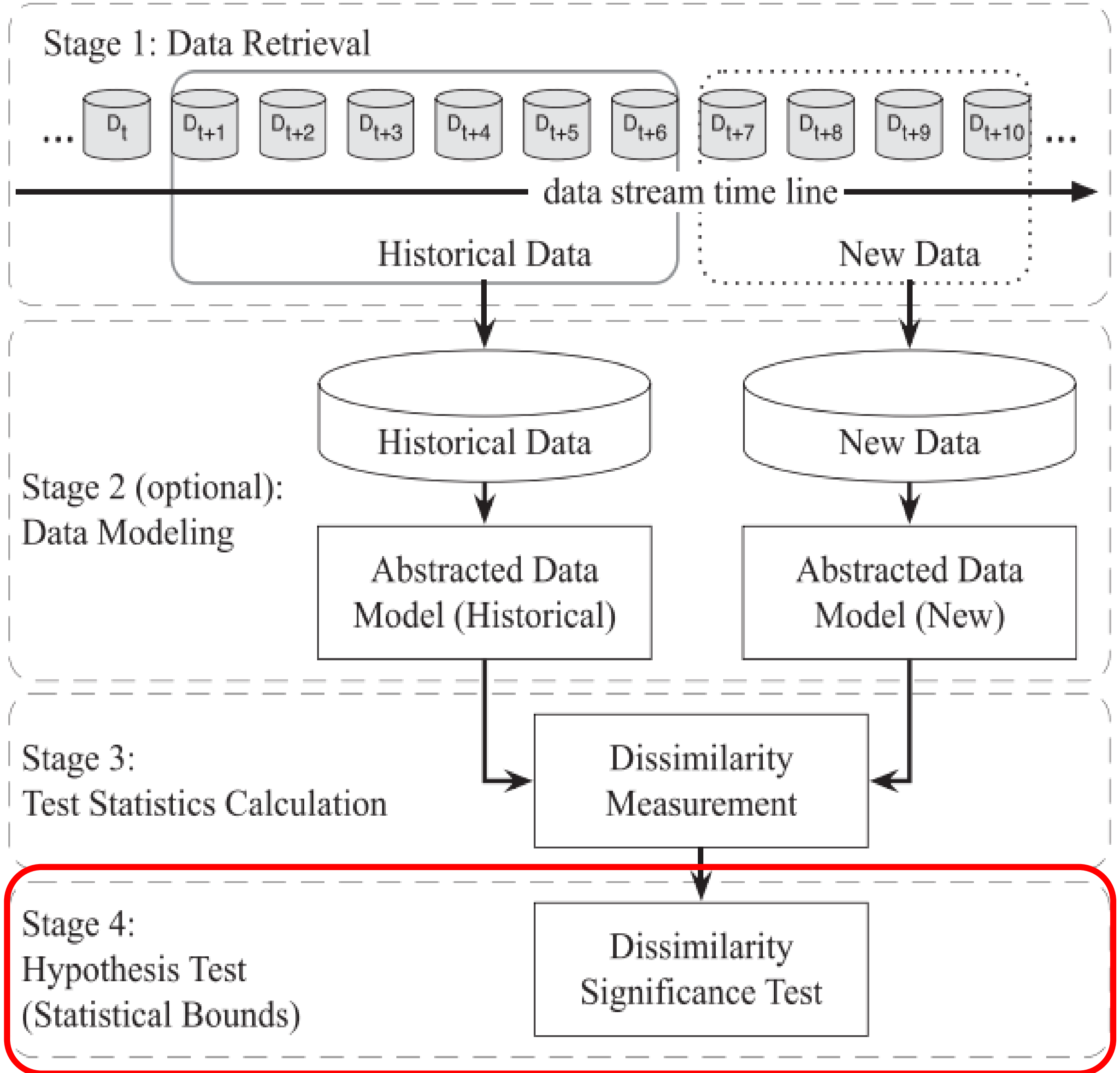
- It is considered to be the most challenging aspect of concept drift detection.

- The problem of how to define an accurate and robust dissimilarity measurement is still an open question.


Stage 1: Data Retrieval
…
D_t
D_{t+1}
D_{t+2}
D_{t+3}
D_{t+4}
D_{t+5}
D_{t+6}
D_{t+7}
D_{t+8}
D_{t+9}
D_{t+10}
…
data stream time line
Historical Data
New Data
Historical Data
New Data
Stage 2 (optional): Data Modeling
Abstracted Data Model (Historical)
Abstracted Data Model (New)
Stage 3: Test Statistics Calculation
Dissimilarity Measurement
Stage 4: Hypothesis Test (Statistical Bounds)
Dissimilarity Significance Test

Concept Drift Detection

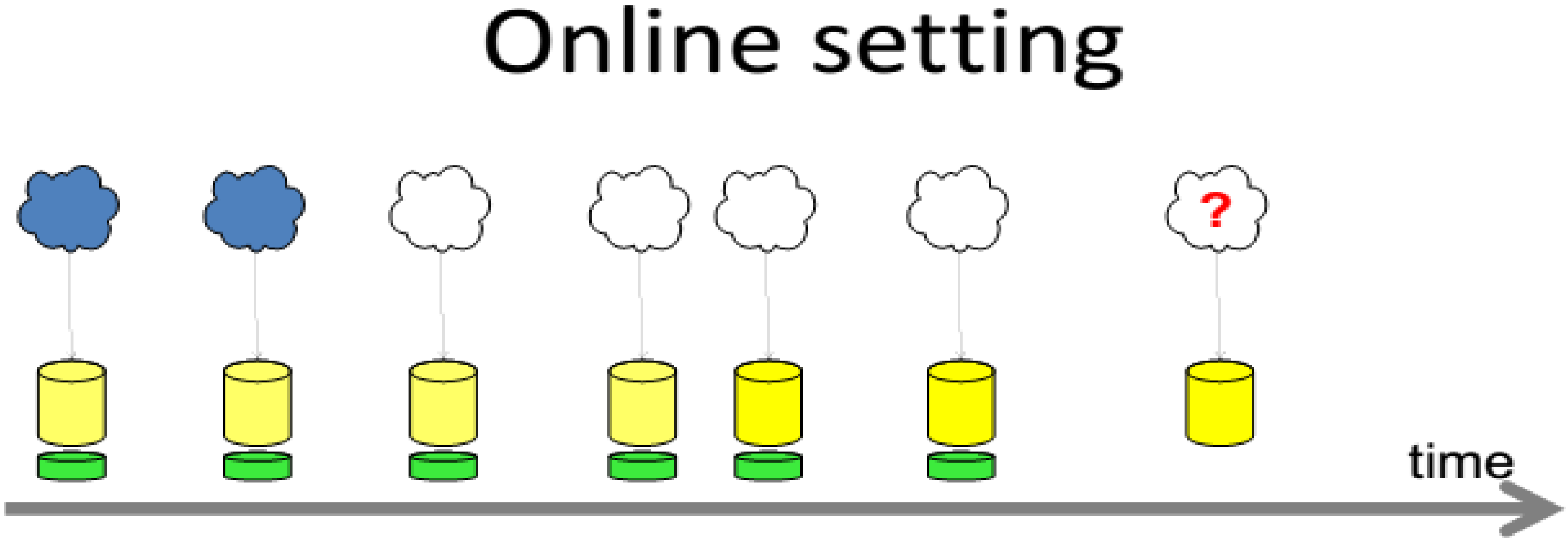
- Stage 4 (Hypothesis Test) uses a specific hypothesis test to evaluate the statistical significance of the change observed in Stage 3.
- They are used to determine drift detection accuracy by proving the statistical bounds of the test statistics proposed in Stage 3.


Stage 1: Data Retrieval
…
D_t
D_{t+1}
D_{t+2}
D_{t+3}
D_{t+4}
D_{t+5}
D_{t+6}
D_{t+7}
D_{t+8}
D_{t+9}
D_{t+10}
…
data stream time line
Historical Data
New Data
Stage 2 (optional): Data Modeling
Historical Data
New Data
Abstracted Data Model (Historical)
Abstracted Data Model (New)
Stage 3: Test Statistics Calculation
Dissimilarity Measurement
Stage 4: Hypothesis Test (Statistical Bounds)
Dissimilarity Significance Test

Drift Detection approaches

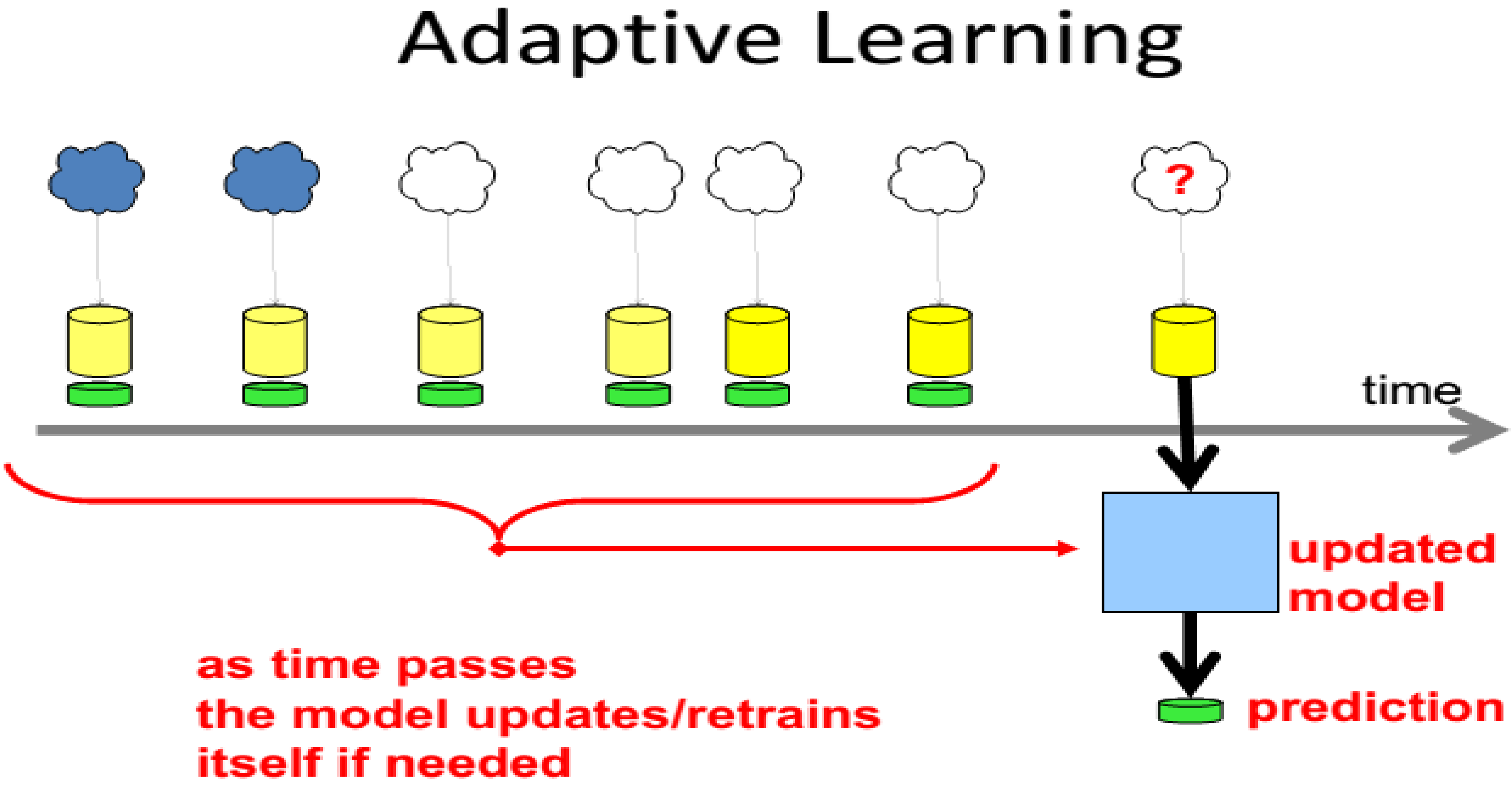
	Explicit drift detection (Supervised)	Implicit drift detection (Unsupervised)
1	Sequential analysis	Novelty detection/ clustering methods
2	Statistical Process Control	Multivariate distribution monitoring
3	Window based distribution monitoring	Model dependent monitoring

The main strategies how to handle concept drift



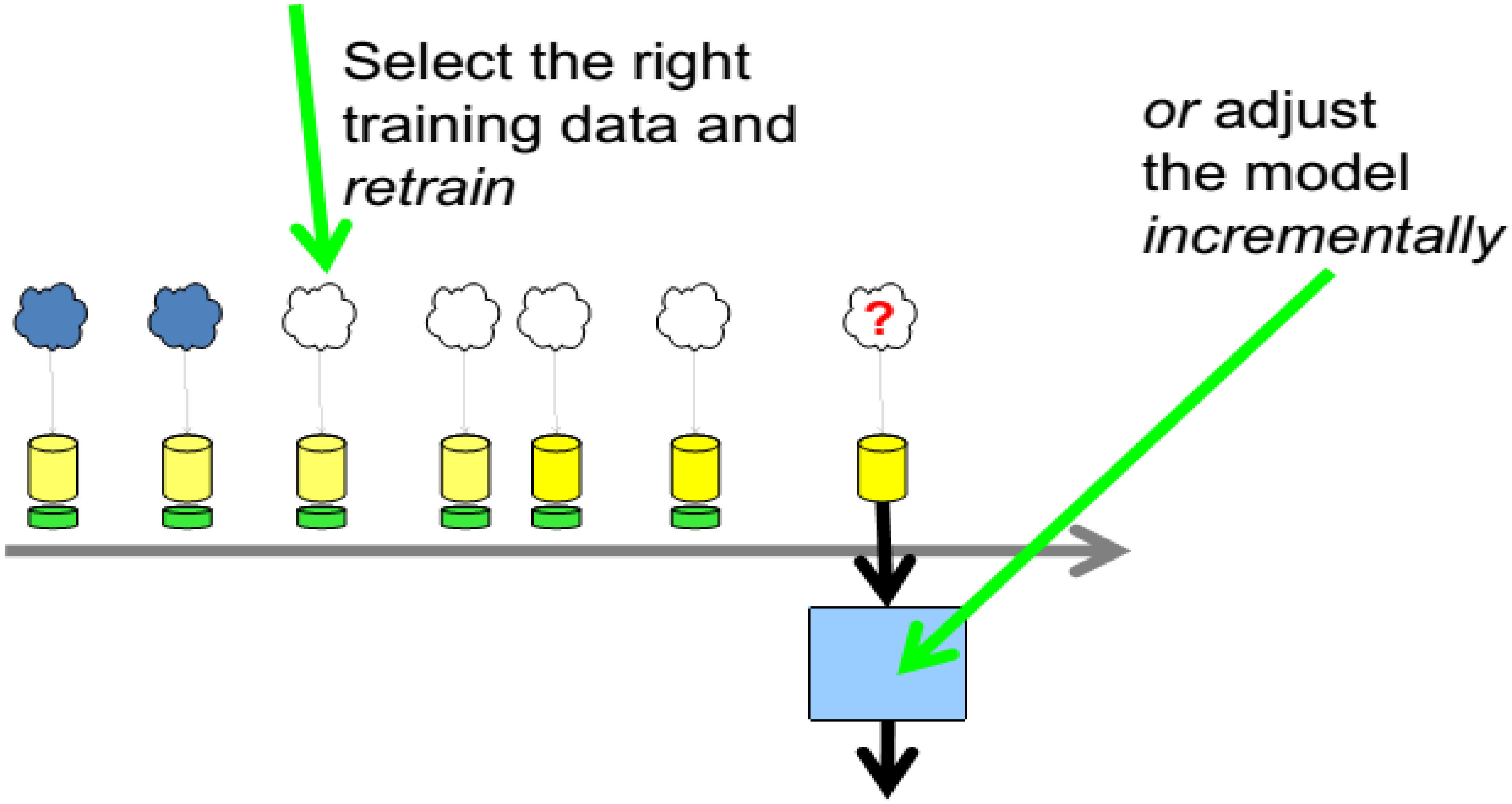
Data arrives in a stream

The main strategies how to handle concept drift



The main strategies how to handle concept drift

How to Adapt

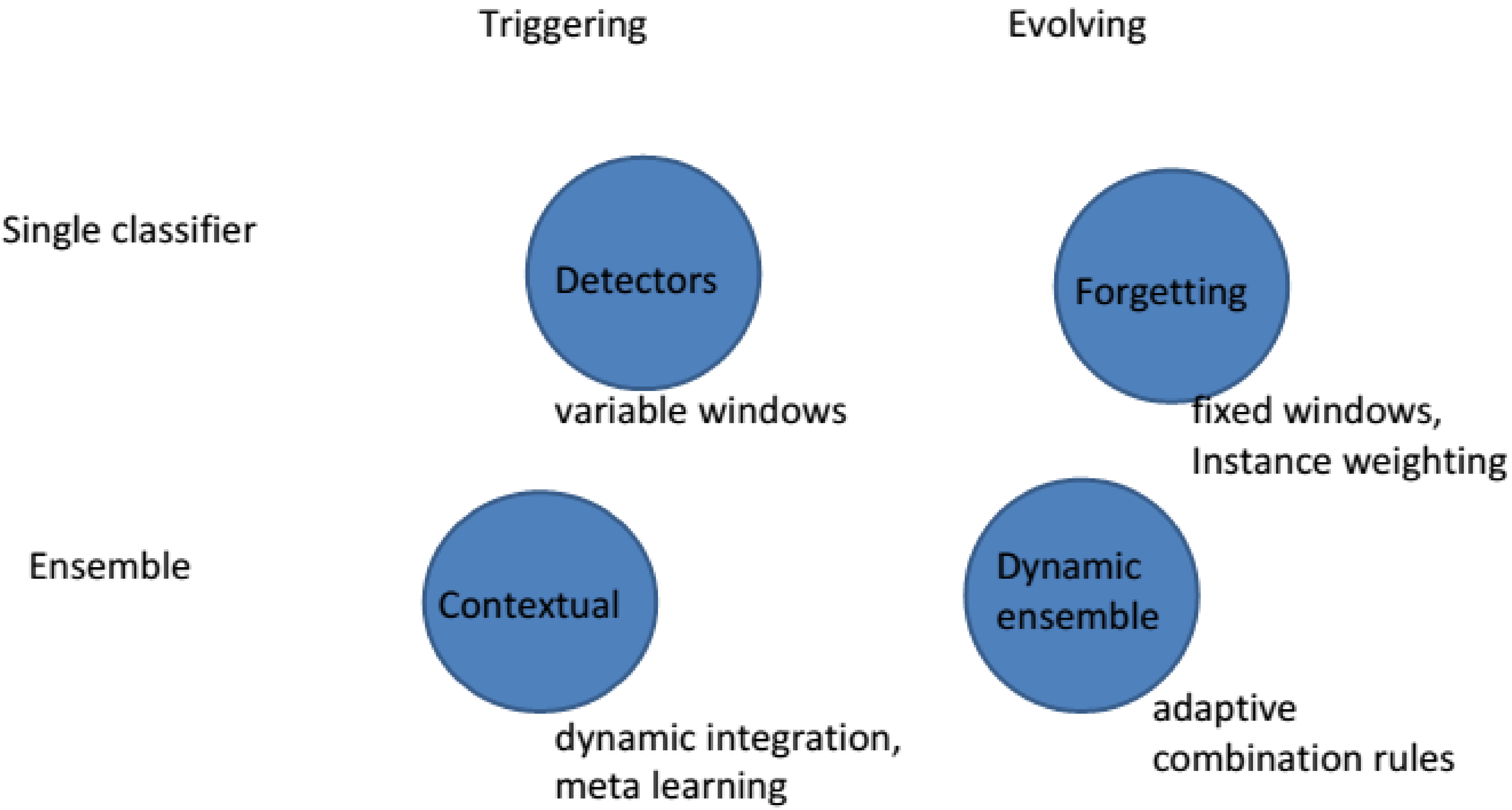


Select the right training data and *retrain*

or adjust the model *incrementally*

The main strategies how to handle concept drift

- Adaptive learning strategies



Triggering

Evolving

Single classifier

Detectors

variable windows

Forgetting

fixed windows,
Instance weighting

Ensemble

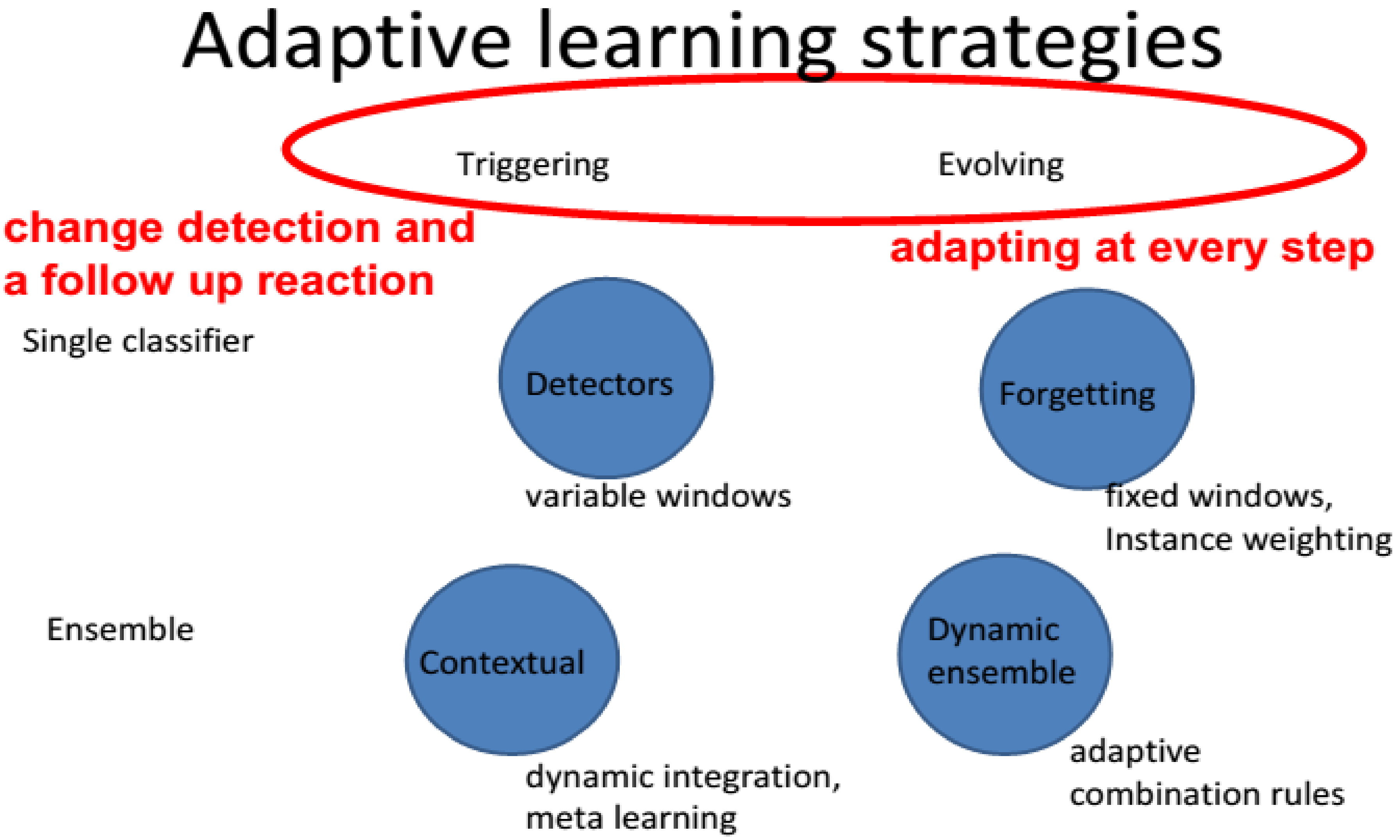
Contextual

dynamic integration,
meta learning

Dynamic ensemble

adaptive
combination rules

The main strategies how to handle concept drift



Adaptive learning strategies

Triggering

Evolving

change detection and a follow up reaction

adapting at every step

Single classifier

Detectors

variable windows

Forgetting

fixed windows, Instance weighting

Ensemble

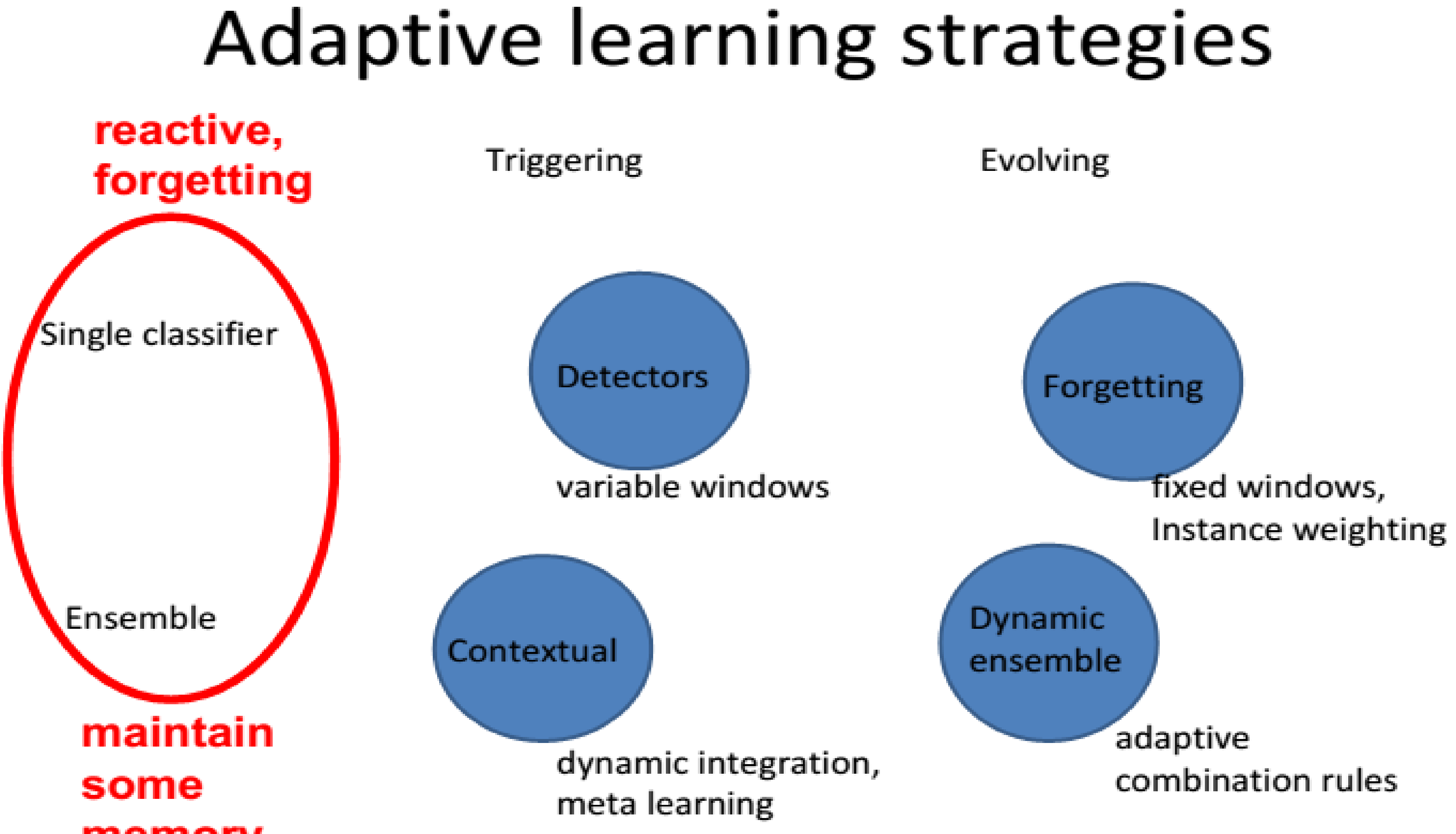
Contextual

dynamic integration, meta learning

Dynamic ensemble

adaptive combination rules

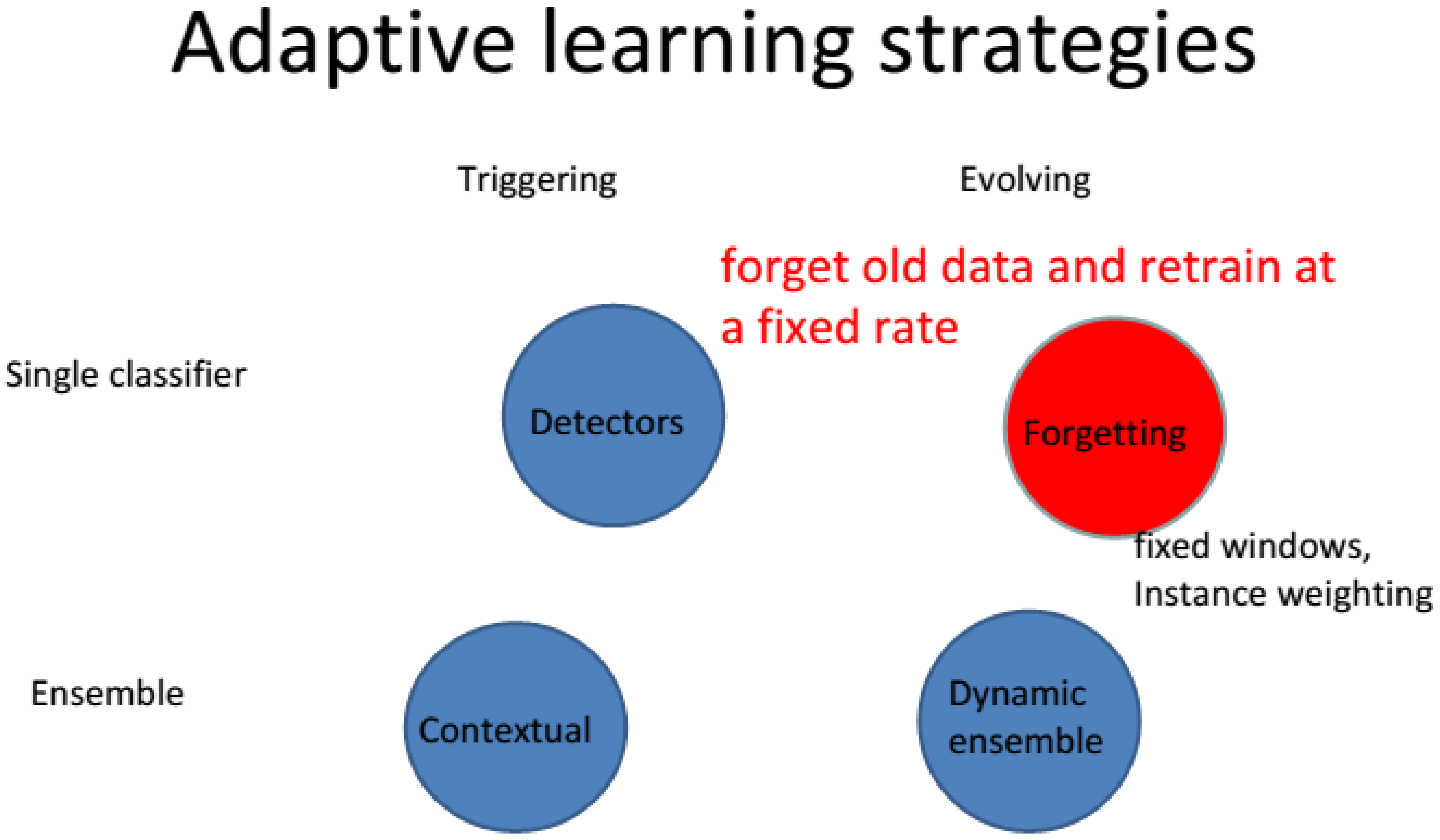
The main strategies how to handle concept drift



reactive, forgetting

maintain some memory

The main strategies how to handle concept drift

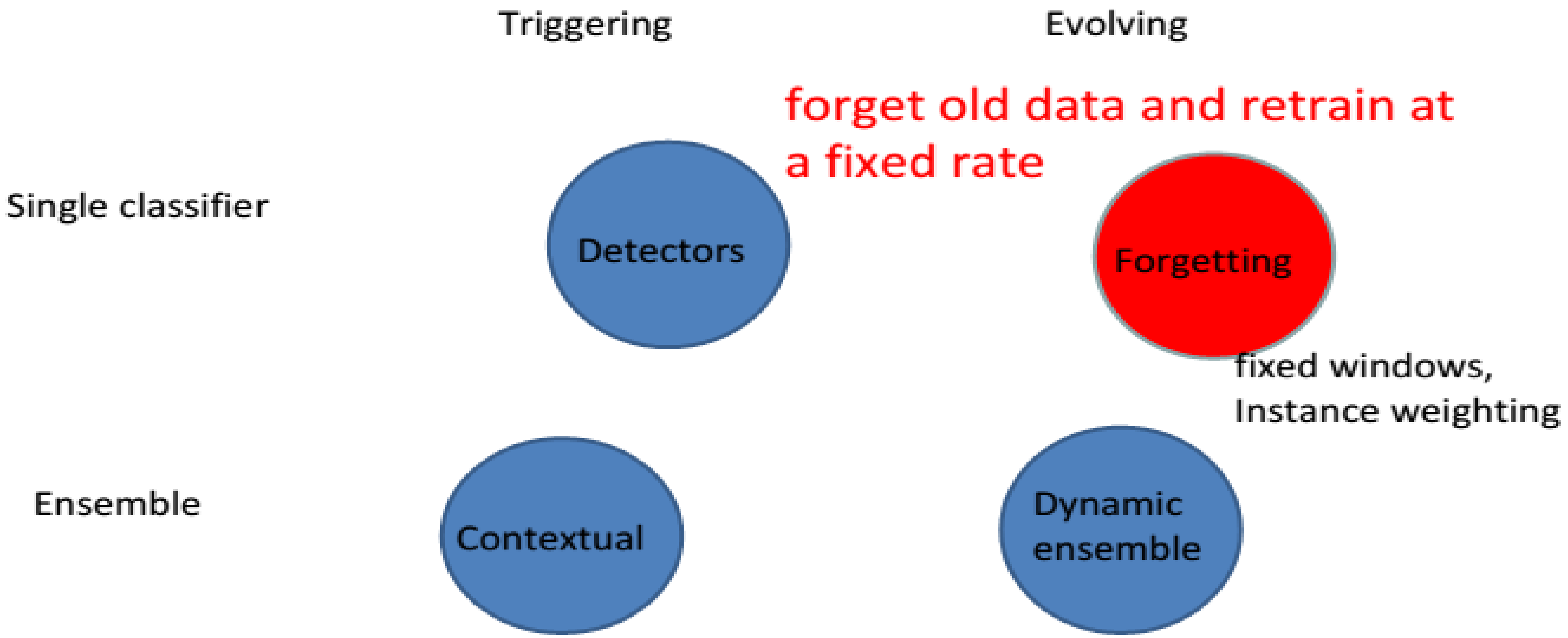


Adaptive learning strategies

forget old data and retrain at a fixed rate

fixed windows, Instance weighting

Adaptive learning strategies



Triggering

Evolving

forget old data and retrain at a fixed rate

Single classifier

Detectors

Forgetting

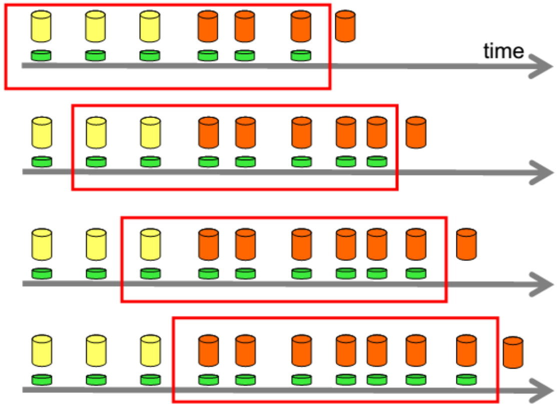
fixed windows, Instance weighting

Ensemble

Contextual

Dynamic ensemble

Fixed Training Window



Adaptive learning strategies

Triggering

Evolving

detect a change and cut

Single classifier

Detectors

Forgetting

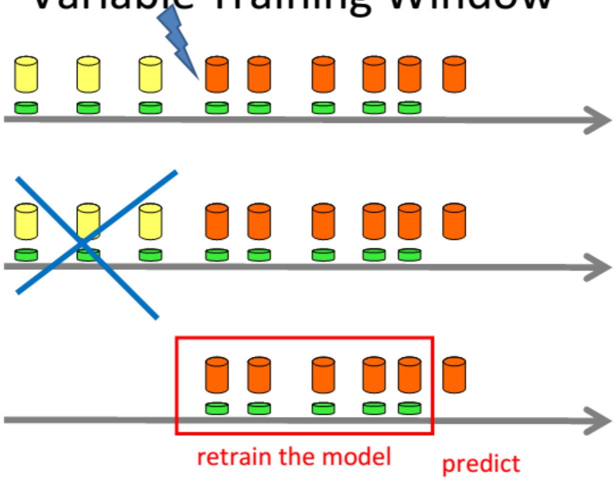
variable windows

Ensemble

Contextual

Dynamic ensemble

Variable Training Window



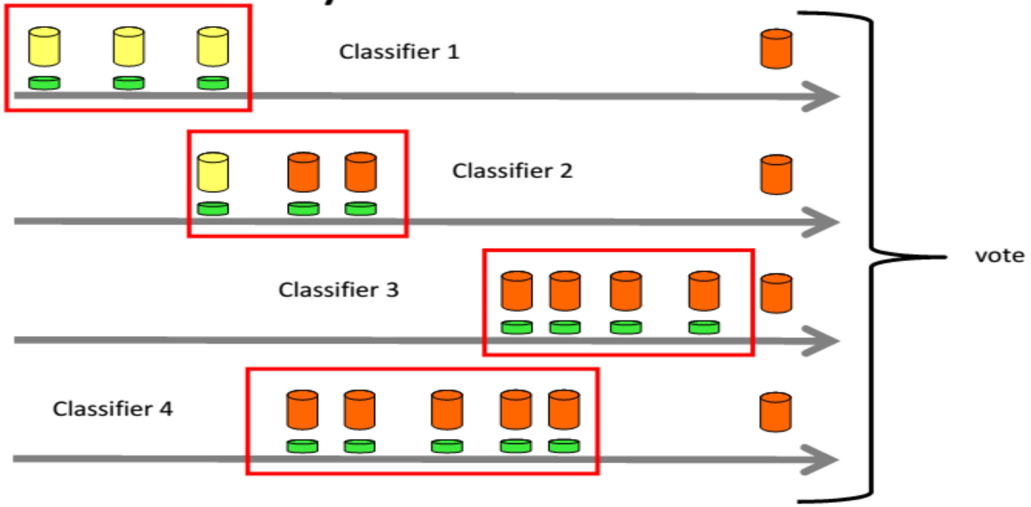
Adaptive learning strategies

	Triggering	Evolving
Single classifier	Detectors	Forgetting
Ensemble	Contextual	Dynamic ensemble

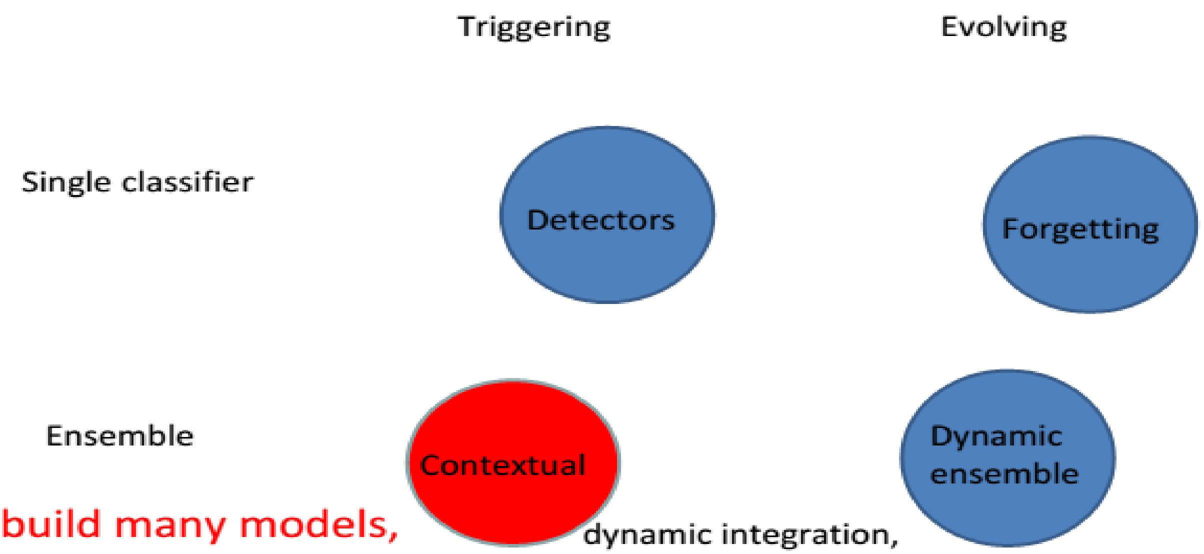
adaptive combination rules

build many models, dynamically combine

Dynamic Ensemble



Adaptive learning strategies



Triggering

Evolving

Single classifier

Detectors

Forgetting

Ensemble

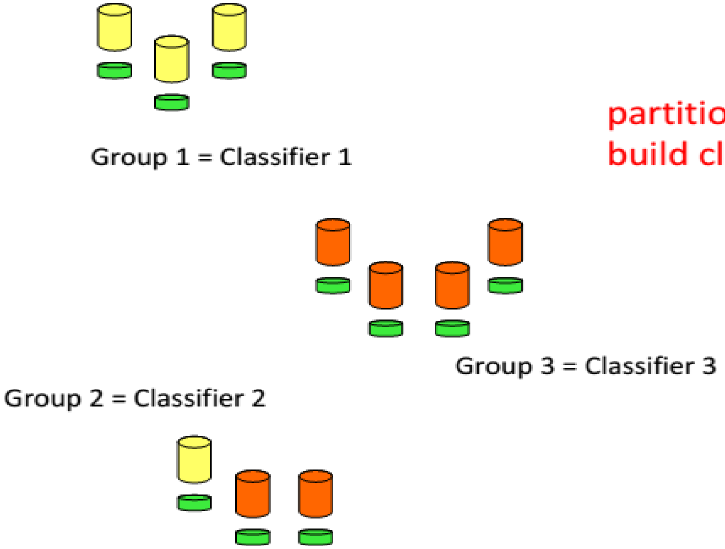
Contextual

Dynamic ensemble

dynamic integration, meta learning

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

Contextual (Meta) Approaches



Group 1 = Classifier 1

Group 3 = Classifier 3

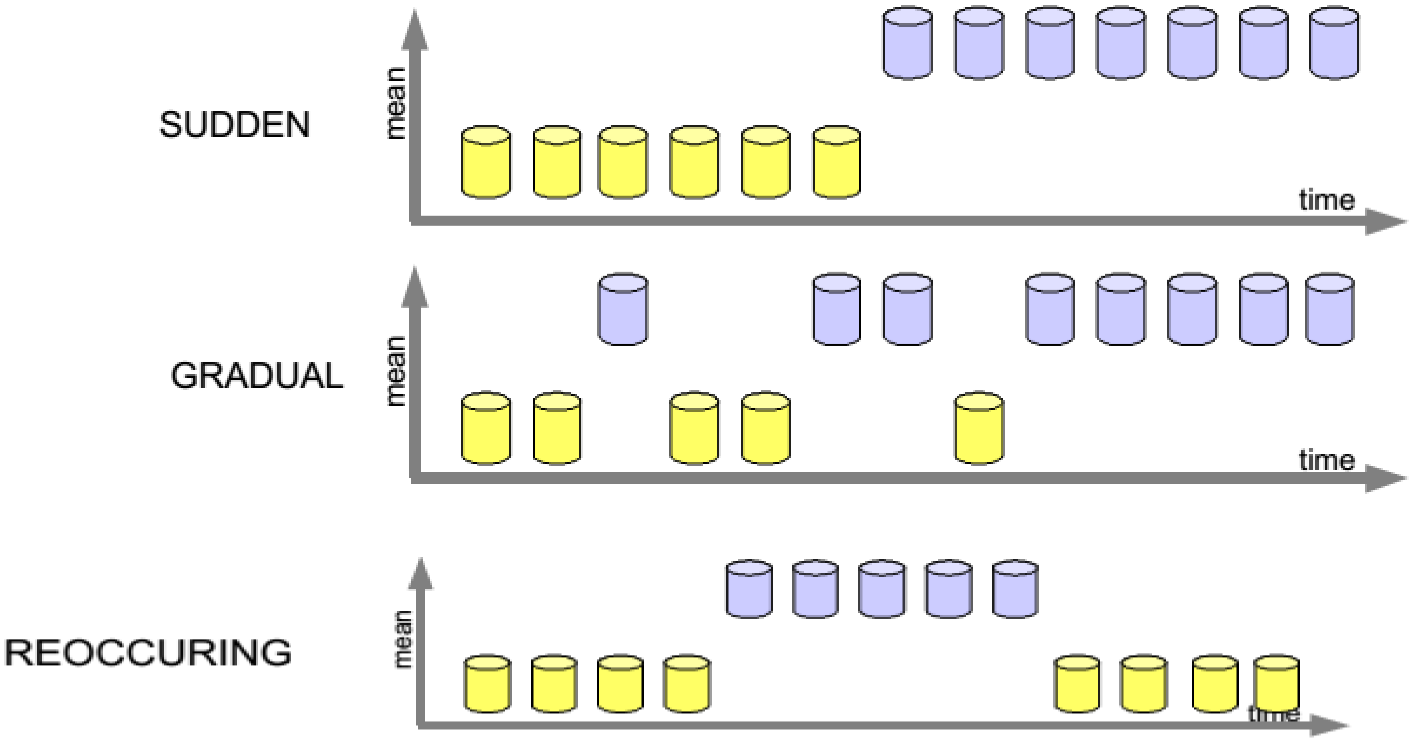
Group 2 = Classifier 2

partition the training data
build classifiers

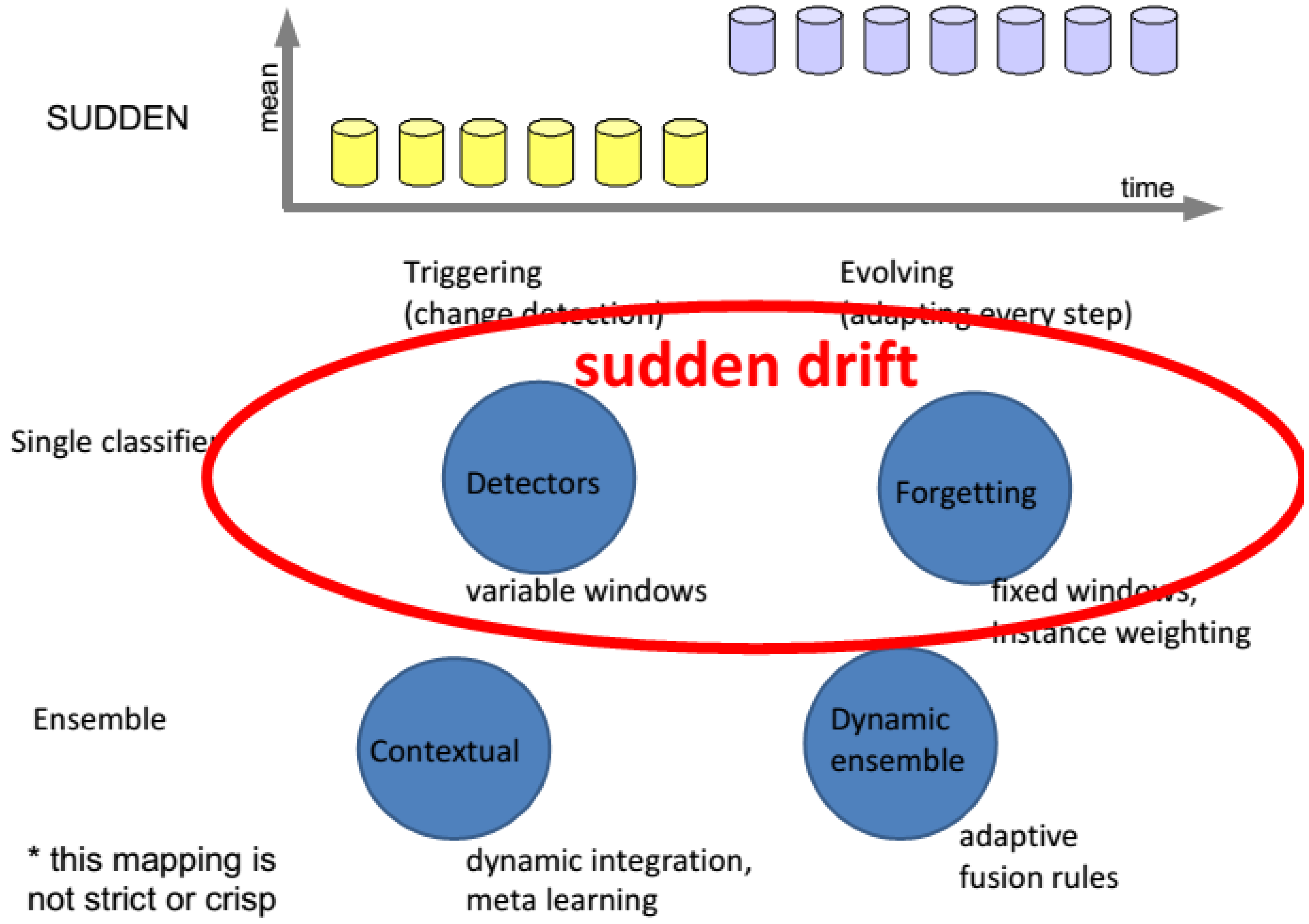
Types of Concept Drifts

- adaptive learning approaches implicitly or explicitly assume some type of change

Types of Changes: speed



Types of Concept Drifts



SUDDEN

Triggering
(change detection)

Evolving
(adapting every step)

sudden drift

Single classifier

Detectors

variable windows

Forgetting

fixed windows,
instance weighting

Ensemble

Contextual

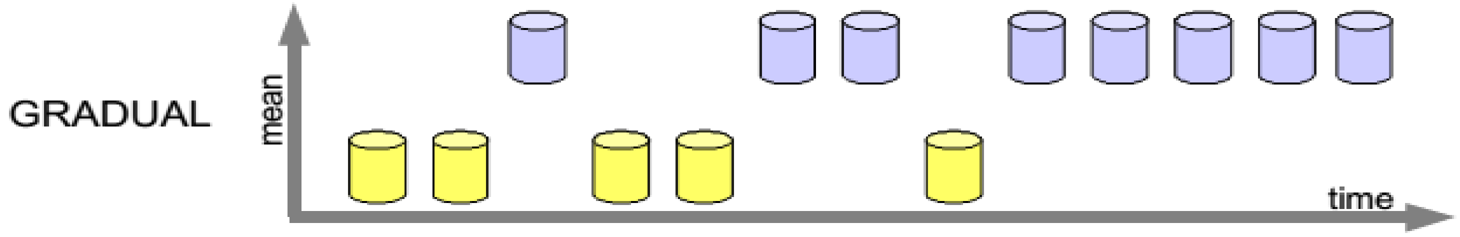
dynamic integration,
meta learning

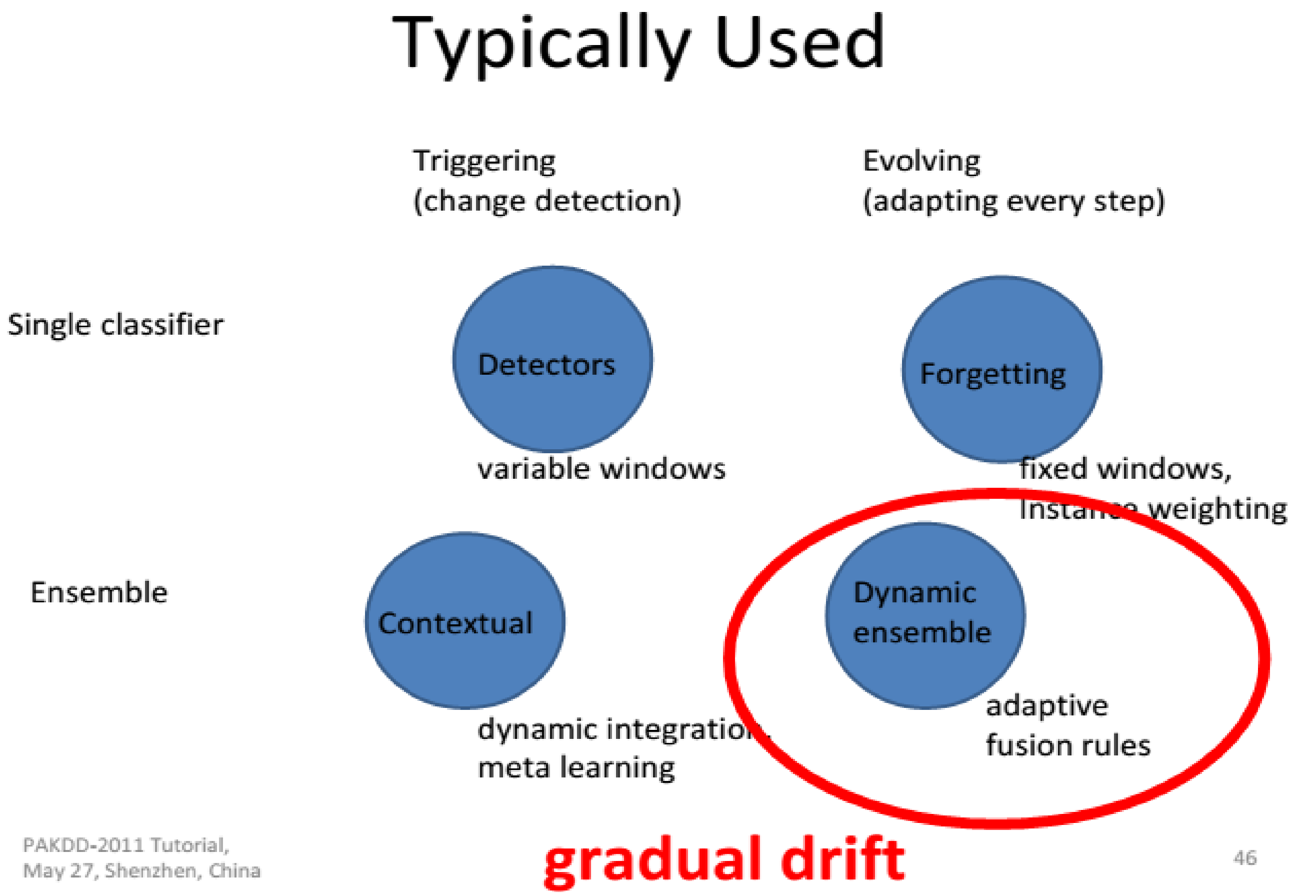
Dynamic
ensemble

adaptive
fusion rules

* this mapping is
not strict or crisp

Types of Concept Drifts





Typically Used

Triggering
(change detection)

Evolving
(adapting every step)

Single classifier

Detectors

variable windows

Forgetting

fixed windows,
Instance weighting

Ensemble

Contextual

dynamic integration,
meta learning

Dynamic ensemble

adaptive
fusion rules

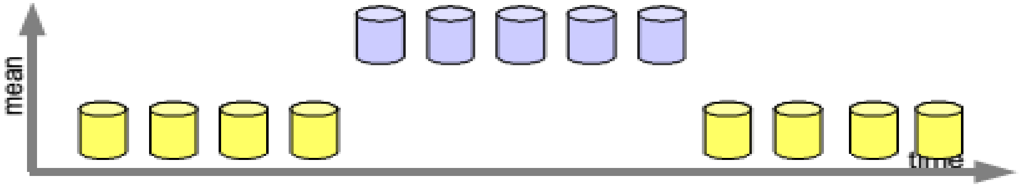
PAKDD-2011 Tutorial,
May 27, Shenzhen, China

46

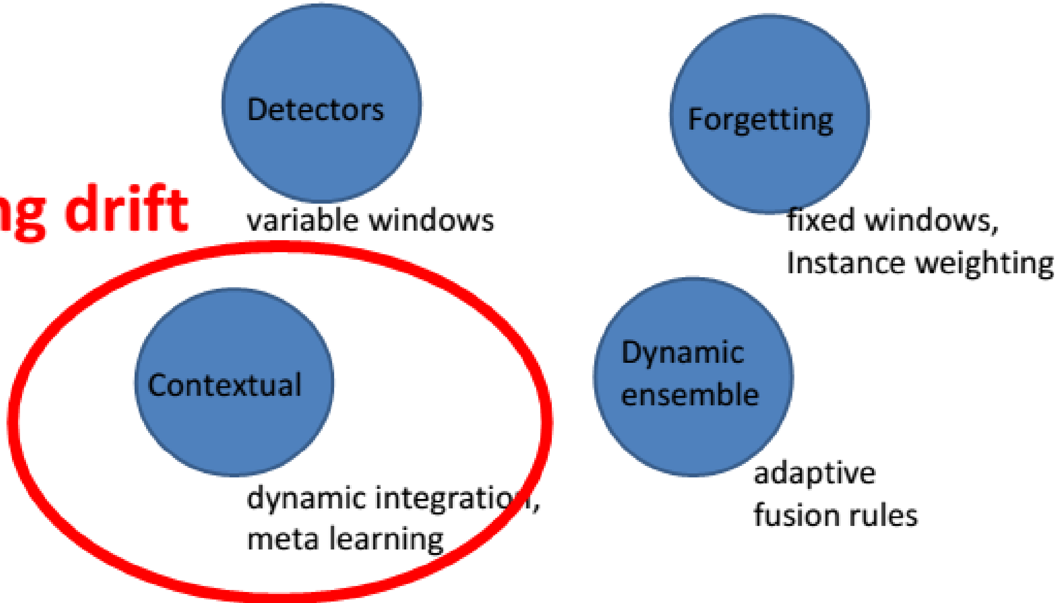
gradual drift

Types of Concept Drifts

REOCCURING



Typically Used



Triggering
(change detection)

Evolving
(adapting every step)

Single classifier

Detectors

variable windows

Forgetting

fixed windows,
Instance weighting

reoccuring drift

Ensemble

Contextual

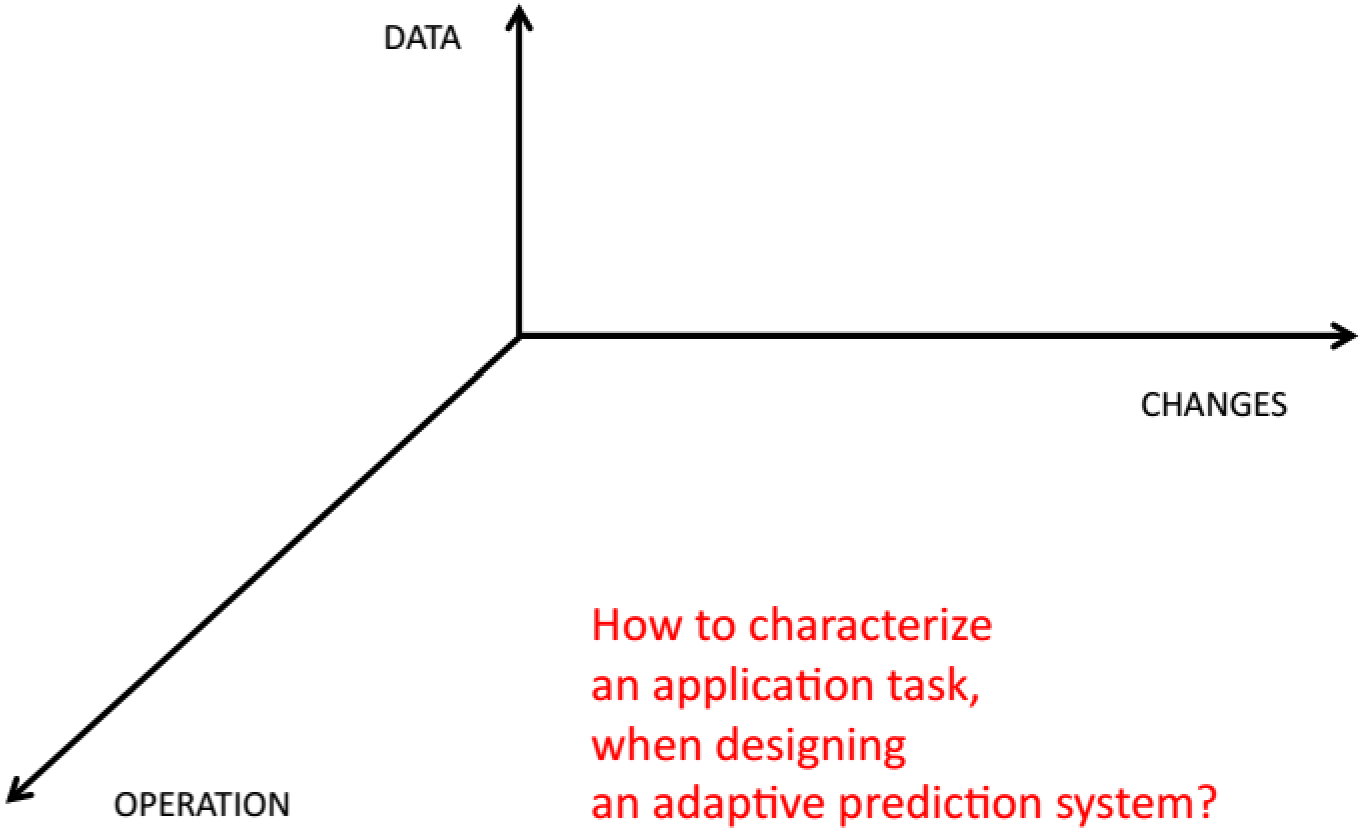
dynamic integration,
meta learning

Dynamic ensemble

adaptive
fusion rules

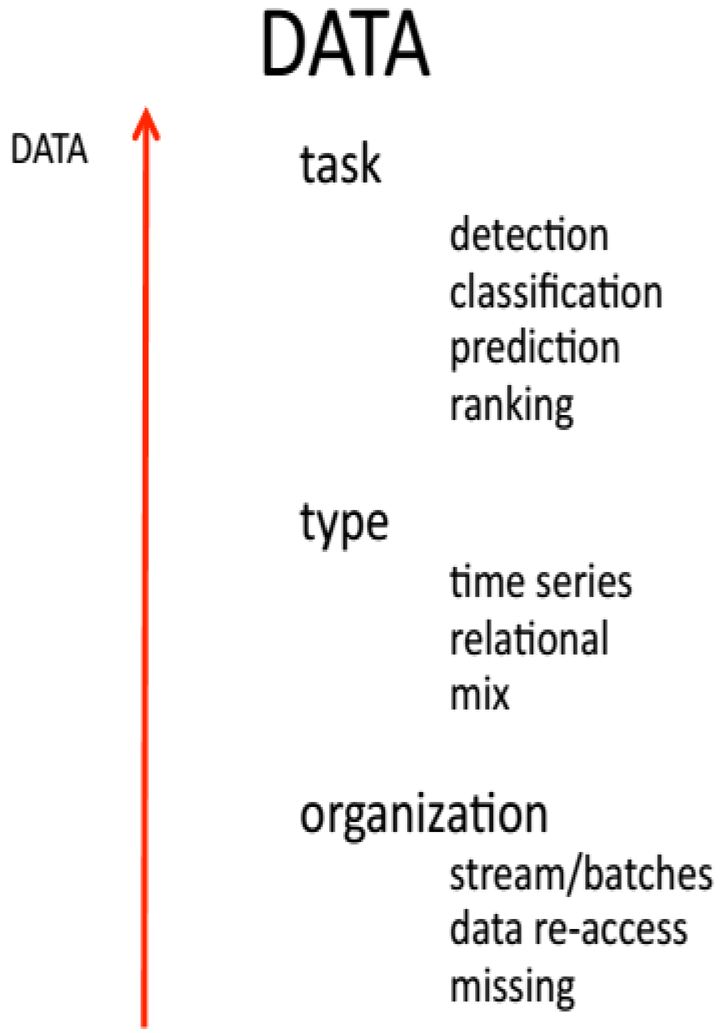
Challenges due to concept drift

Application tasks



How to characterize
an application task,
when designing
an adaptive prediction system?

Challenges due to concept drift



DATA

DATA

task
- detection
- classification
- prediction
- ranking

type
- time series
- relational
- mix

organization
- stream/batches
- data re-access
- missing

Challenges due to concept drift

CHANGE

DATA
CHANGES

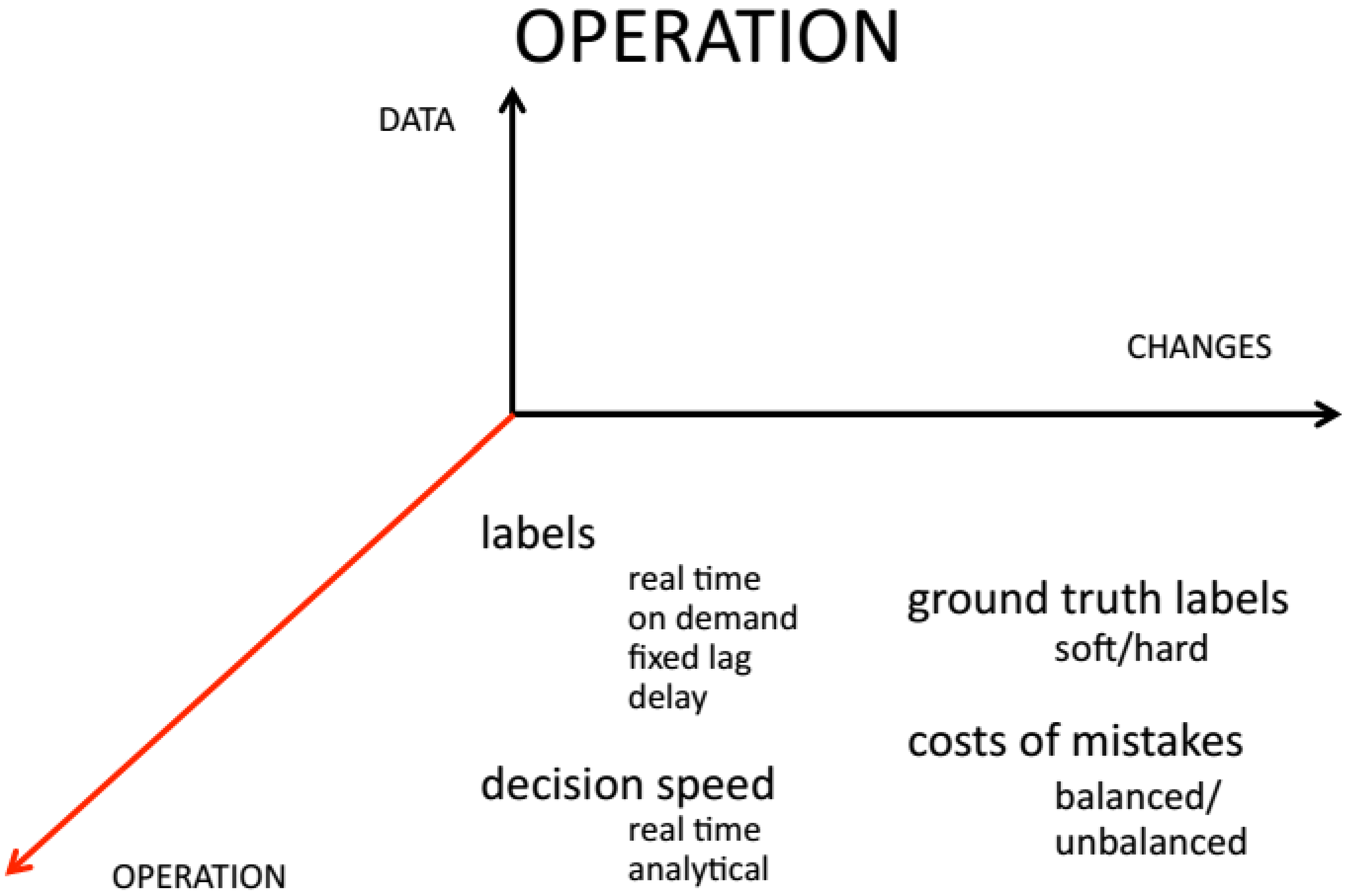
change type
- sudden
- gradual
- incremental
- reoccurring

change source
- adversary
- interests
- population
- complexity

expectations about changes
- unpredictable
- predictable
- identifiable

expectations about desired action
- keep the model uptodate
- detect the change
- identify/locate the change
- explain the change

Challenges due to concept drift



OPERATION

DATA

CHANGES

OPERATION

labels

real time
on demand
fixed lag
delay

decision speed

real time
analytical

ground truth labels

soft/hard

costs of mistakes

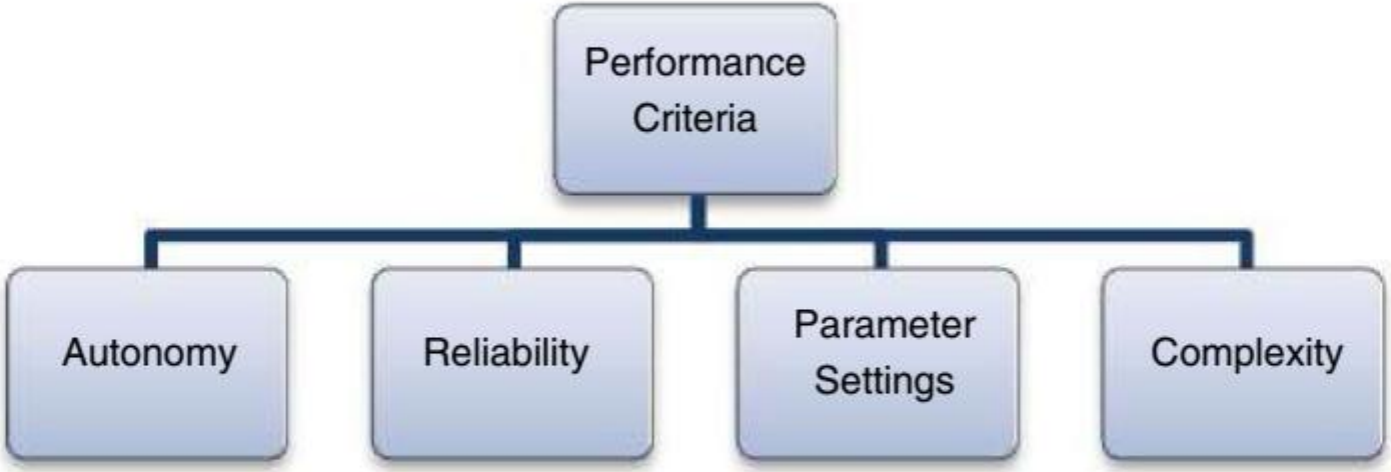
balanced/
unbalanced

Evaluation Systems

- Several criteria:
  - Time → seconds
  - Memory → RAM/hour
  - Generalizability of the model → % success
  - Detecting concept drift → detected drifts, false positives and false negatives

Evaluation Systems

- Taxonomy of performance criteria for handling concept drift
- According to the requirements of the real world applications, the performance criteria can concern:
  — Autonomy: the level of human involvement,
  — Reliability: the accuracy of drift information,
  — Parameter settings: the availability of *a priori* knowledge,
  — Complexity: the time and memory consumption



Explicit concept drift detection methodologies

- **Sequential analysis methodologies-**
  - Continuously monitor the sequence of performance metrics , such as
    - Accuracy
    - F-measure
    - precision and recall;
  - to signal a change, in the event of a significant drop in these values.
- Methodologies comes under the Sequential analysis-
  - CUSUM (Cumulative Sum approach)
  - PHT (PageHinckley Test)

Sequential analysis methodologies

- **CUSUM(Cumulative Sum approach)- This approach signals an alarm when the mean of the sequence significantly deviates from 0.**
- **The CUSUM test monitors a metric M, at time t, on an incoming sample’s performance εt , using parameters v for acceptable deviation and θ for the change threshold as given in the equation.**

$$M_0 = 0; \quad M_t = max(0, M_{t-1} + \epsilon_t - v)$$
$$if \quad M_t > \theta \quad then \quad 'alarm' \quad and \quad M_t = 0$$

- **Max function is used to test changes in positive direction. For reverse effect a min function can be used.**
- Memory-less and can be used incrementally.

Sequential analysis methodologies

- PageHinckley Test (PHT) is a variant of CUSUM approach.
- PHT monitor the metric as an accumulated difference between its mean and current values, as shown below.

$$M_0 = 0; \quad M_t = M_{t-1} + (\epsilon_t - v); \quad M_{Ref} = \min(V)$$
$$if \quad M_t - M_{Ref} > \theta \quad then \quad 'alarm' \quad and \quad M_t = 0$$

- Where, M_0 is the initial metric at time t = 0. M_t is the current metric computed far (M_t−1) and the sample’s performance at time t = εt and v denotes acceptable deviation from mean and θ is the change detection threshold.

Statistical Process Control based methodologies-

- Monitor the online trace of error rates, and detects deviations based on ideas taken from control charts.
- A significantly increased error rate violates the model and as such is assumed to be a result of concept drift.
- Methodologies under this category are-

  - DDM (Drift Detection Method)

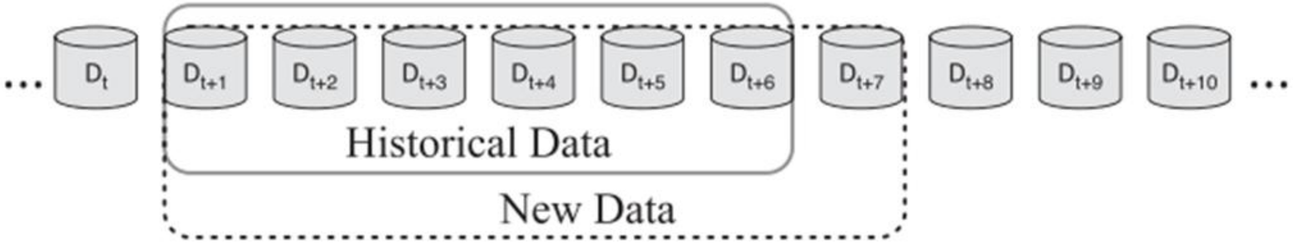
  - EDDM (Early Drift Detection Method)

  - STEPD (Statistical Test of Equal Proportion Distribution)

  - EWMA (Exponentially Weighted Moving Average)

Statistical Process Control based methodologies

- DDM(Drift Detection Method)-
  - Monitors the probability of error at time t as p_t and the standard deviation as $s_t=\sqrt{p_t}(1-p_t\ )/i$.
  - When, $p_t + s_t$ reaches its minimum value, the corresponding values are stored in p_{min} and s_{min}.
  - A warning is signaled when $p_t + s_t \geq p_{min} + 2 * s_{min}$.
  - and a drift is signaled when $p_t + s_t \geq p_{min} + 3 * s_{min}$.



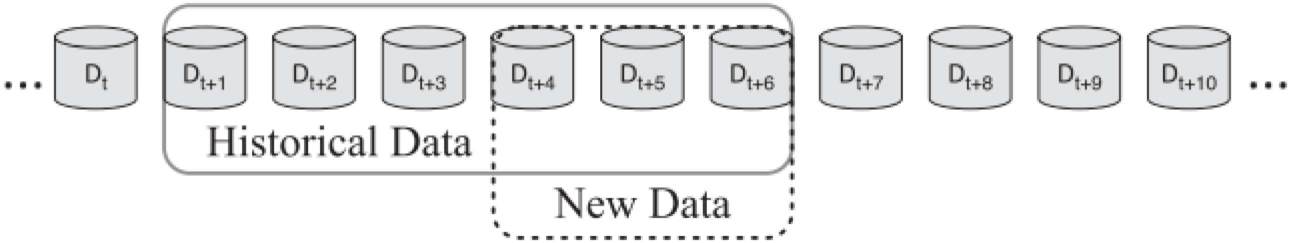
Landmark time window for drift detection. The starting point of the window is fixed, while the end point of the window will be extended after a new data instance has been received.

Statistical Process Control based methodologies

- EDDM(Early Drift Detection Methodology)
  - An extension of DDM, and was made suitable for slow moving gradual drifts, where DDM previously failed.
  - EDDM monitors the number of samples between two classification errors, as a metric to be tracked online for drift detection.
  - Based on the model, it was assumed that, in stationary environments, the distance (in number of samples) between two subsequent errors would increase.
  - A violation of these condition was seen to be indicative of drift.

Statistical Process Control based methodologies

- STEPD(Statistical Test of Equal Proportions)
  - Computes the accuracy of a chunk C of recent samples and compares it with the overall accuracy from the beginning of the stream, using a chi-squares test to check for deviation.



Two time windows for concept drift detection. The new data window has to be defined by the user.

Statistical Process Control based methodologies

- EWMA(Exponentially Weighted Moving Average)-
  - An incremental approach proposed in,where the EWMA was used to signal deviation in the average error rate, in terms of the number of standard deviations from the mean.
  - The metric M (here, error rate) at time t is updated as per the equation-

$$M_0 = \mu_0; \quad M_t = \lambda * M_{t-1} + (1-\lambda) * \epsilon_t$$

$$if \quad M_t - \mu_0 > \theta * \sigma_0 \quad then \quad 'alarm'$$

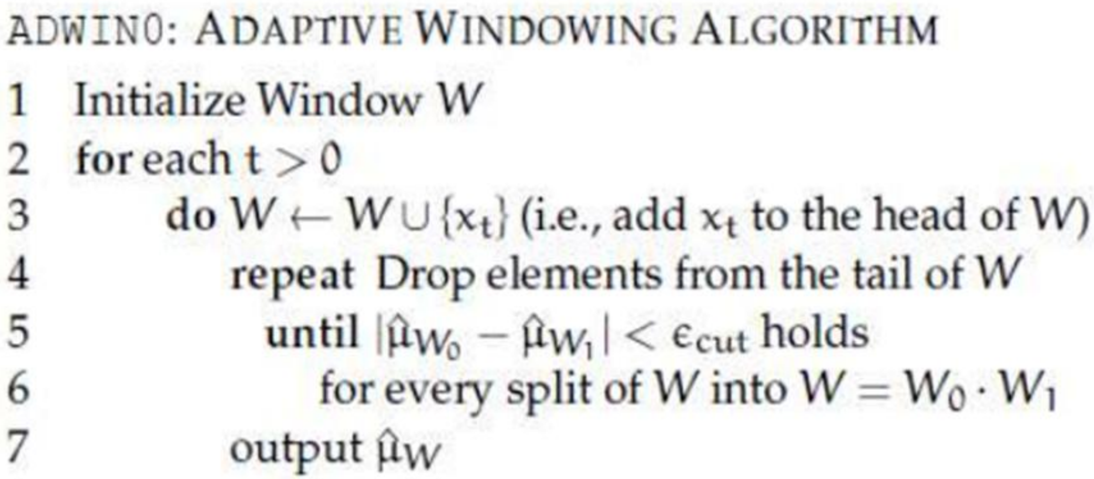
  - Where, μ_0 and σ_0 are mean and standard deviation obtained from the training data ,error rate at time t is given by ε_t ,θ is the acceptable deviation from the mean and λ is the forgetting factor which controls the effect of previous data on the current sample.

Window based distribution monitoring methodologies

- Window based approaches use a chunk based or sliding window approach over the recent samples, to detect changes.
- Deviations are computed by comparing the current chunk's distribution to a reference distribution, obtained at the start of the stream, from the training dataset.
- These approaches provide precise localization of change point, and are robust to noise and transient changes.
- Extra memory is required to store these two distributions over time.

Window based distribution monitoring methodologies

- ADWIN(Adaptive Windowing)
  - This algorithm of uses a variable length sliding window, whose length is computed online according to the observed changes.
  - Whenever two large enough sub windows of the current chunk exhibit distinct averages of the performance metric, a drift is detected.
  - Hoeffding bounds are used to determine optimal change threshold and window parameters.



ADWIN0: ADAPTIVE WINDOWING ALGORITHM

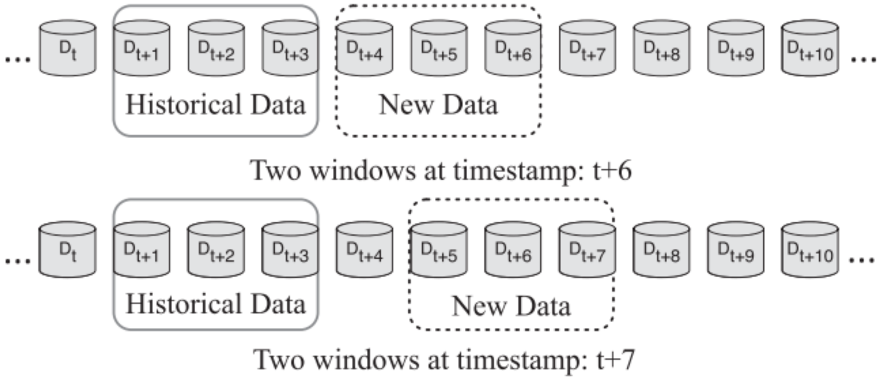
```
Initialize Window $W$
for each $t > 0$
    do $W \leftarrow W \cup \{x_t\}$ (i.e., add $x_t$ to the head of $W$)
        repeat Drop elements from the tail of $W$
          until $|\hat{\mu}_{W_0} - \hat{\mu}_{W_1}| < \epsilon_{cut}$ holds
              for every split of $W$ into $W = W_0 \cdot W_1$
        output $\hat{\mu}_W$
```

Window based distribution monitoring methodologies

- **DoD (Degree of Drift)**
  - Detects drifts by computing a distance map of all samples in the current chunk and their nearest neighbors from the previous chunk.
  - If the distance increases more than a parameter θ, a drift is signaled.
  - Drift is managed by replacing the stable model with the reactive one and setting the circular disagreement list to all zeros.

Implicit drift detection methodologies

- **Novelty detection / Clustering based methods**
  - Capable of identifying uncertain suspicious samples, which need further evaluation.
  - An additional 'Unknown' class label to indicate that these suspicious samples do not fit the existing view of the data.
  - Clustering and outlier based approaches are popular for detecting novel patterns, as they summarize current data and can use dissimilarity metrics to identify new samples.



Two sliding time windows, of fixed size. The historical data window will be fixed while the new data window will keep moving.

Novelty detection / Clustering based methods

OLINDDA(OnLIne Novelty and Drift Detection Algorithm)

- Uses K-means data clustering to continuously monitor and adapt to emerging data distribution.
- Unknown samples are stored in a short term memory queue, and are periodically clustered and then either merged with existing similar cluster profiles or added as a novel profile to the pool of clusters.

Novelty detection / Clustering based methods

- All novelty detection techniques rely on clustering to recognize new regions of space, which were previously unseen.
- They suffer from the curse of dimensionality, being distance dependent, and also the problem of dealing with binary data spaces.
- Additionally, they are suitable to detect only specific type of cluster-able drifts.

Multivariate distribution monitoring methods

- These approaches are primarily chunk based and store summarized information of the training data chunk (as histograms of binned values), as the reference distribution, to monitor changes in the current data chunk.
- Hellinger distance and KL-divergence are commonly used to measure differences between the two chunk distributions and to signal drift in the event of a significant change.

Multivariate distribution monitoring methods

- **Change of Concept(CoC)**

- This technique considers each feature as an independent stream of data and monitors correlation using Pearson correlation between the current chunk and the reference training chunk.
- Change in the average correlation over the features is used as a signal of change.

Multivariate distribution monitoring methods

- HDDDM(Hellinger Distance Drift Detection Methodology)-
  - A non parametric chunk based approach which uses Hellinger distance to measure change in distribution, over time.
  - An increased Hellinger distance, between the current stream chunk and a training reference chunk, is used to signal drift.
  - The Hellinger distance (HD) between the reference chunk P and the current chunk Q is computed as-

$$HD(P,Q) = \frac{1}{d}\sum_{k=1}^{d}\sqrt{\sum_{i=1}^{b}\left(\sqrt{\frac{P_{i,k}}{\sum_{j=1}^{b}P_{j,k}}}-\sqrt{\frac{Q_{i,k}}{\sum_{j=1}^{b}Q_{j,k}}}\right)^{2}}$$

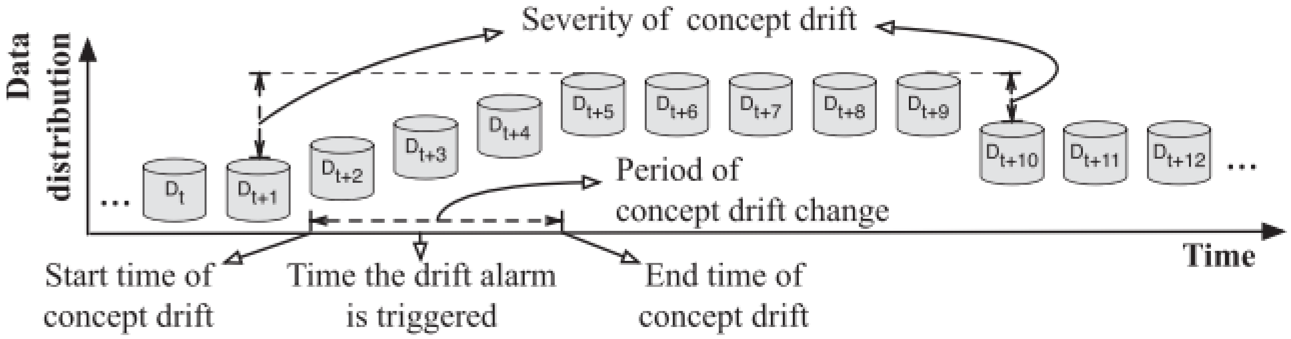
  - Here, N is the number of samples in the chunk, d is the data dimensionality and b is the number of bins (b= √ N), per feature.

Model dependent drift detection methodologies

- The model dependent approaches directly consider the classification process by tracking the posterior probability estimates of classifiers, to detect drift.
- By monitoring the posterior probability estimates, the drift detection task is reduced to that of monitoring a univariate stream of values, making the process computationally efficient.
- Following are the techniques used in this –
  - A-distance: A reduced false positive rate was obtained by tracking the 'A-distance', which was proposed as a measure of histogram difference obtained by binning the margin distribution of samples, between the reference and current margin samples.

Concept Drift Understanding

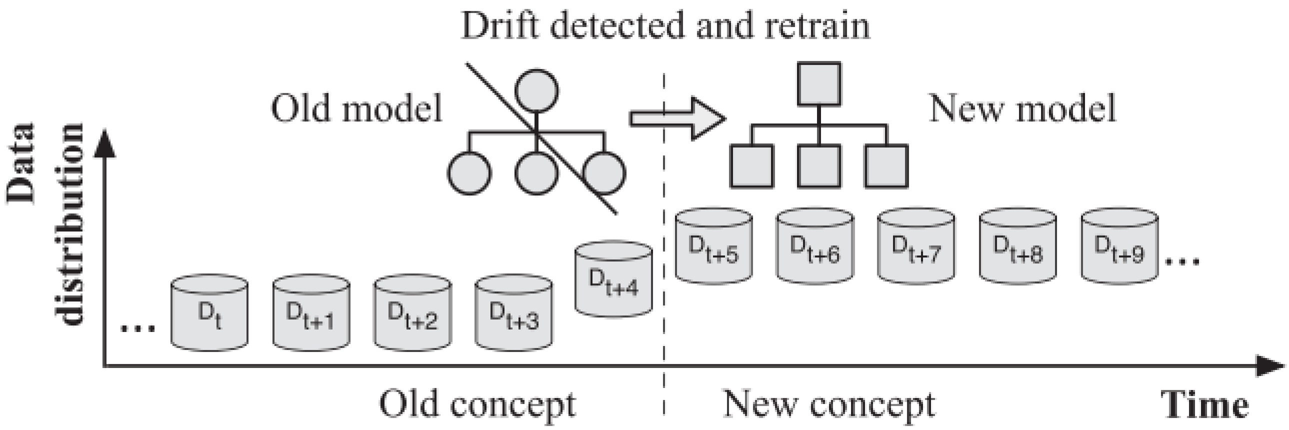
- Drift understanding refers to retrieving concept drift information about
  - "When" (the time at which the concept drift occurs and how long the drift lasts),
  - "How" (the severity /degree of concept drift), and
  - "Where" (the drift regions of concept drift).
- This status information is the output of the drift detection algorithms, and is used as input for drift adaptation.



Drift Adaptation Techniques

- It focuses on strategies for updating existing learning models according to the drift.
- There are three main groups of drift adaptation methods:
  - simple retraining
  - ensemble retraining
  - model adjusting

Training New Models for Global Drift



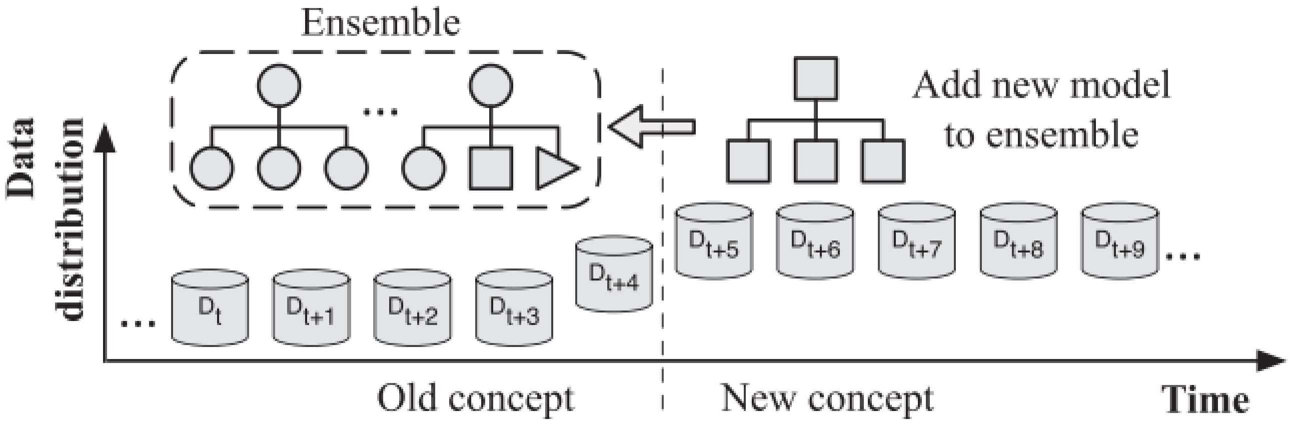
- A new model is trained with latest data to replace the old model when a concept drift is detected.

Training New Models for Global Drift

- Paired Learners follows this strategy and uses two learners:
  - The **stable learner** and the **reactive learner.**
  - If the stable learner frequently misclassifies instances that the reactive learner correctly classifies, a new concept is detected and the stable learner will be replaced with the reactive learner.
- This method is simple to understand and easy to implement, and can be applied at any point in the data stream.

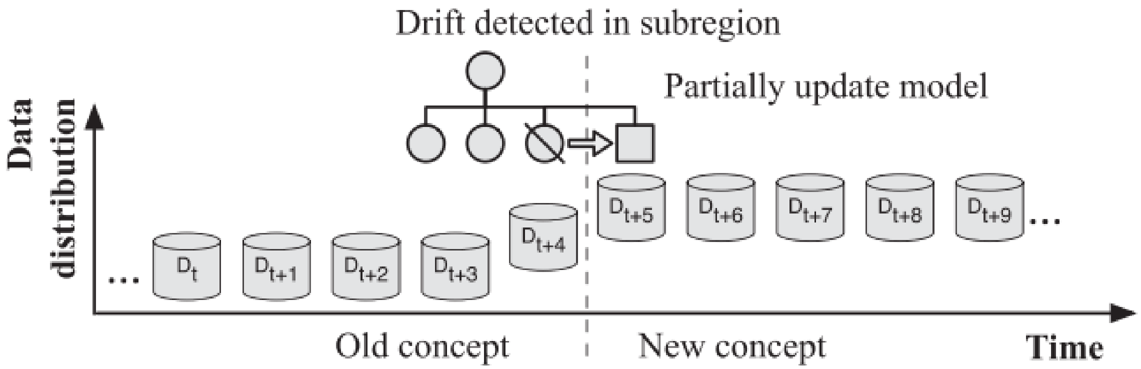
Model Ensemble for Recurring Drift

- In the case of recurring concept drift, preserving and reusing old models can save significant effort to retrain a new model for recurring concepts.
- This is the core idea of using ensemble methods to handle concept drift



Adjusting Existing Models for Regional Drift

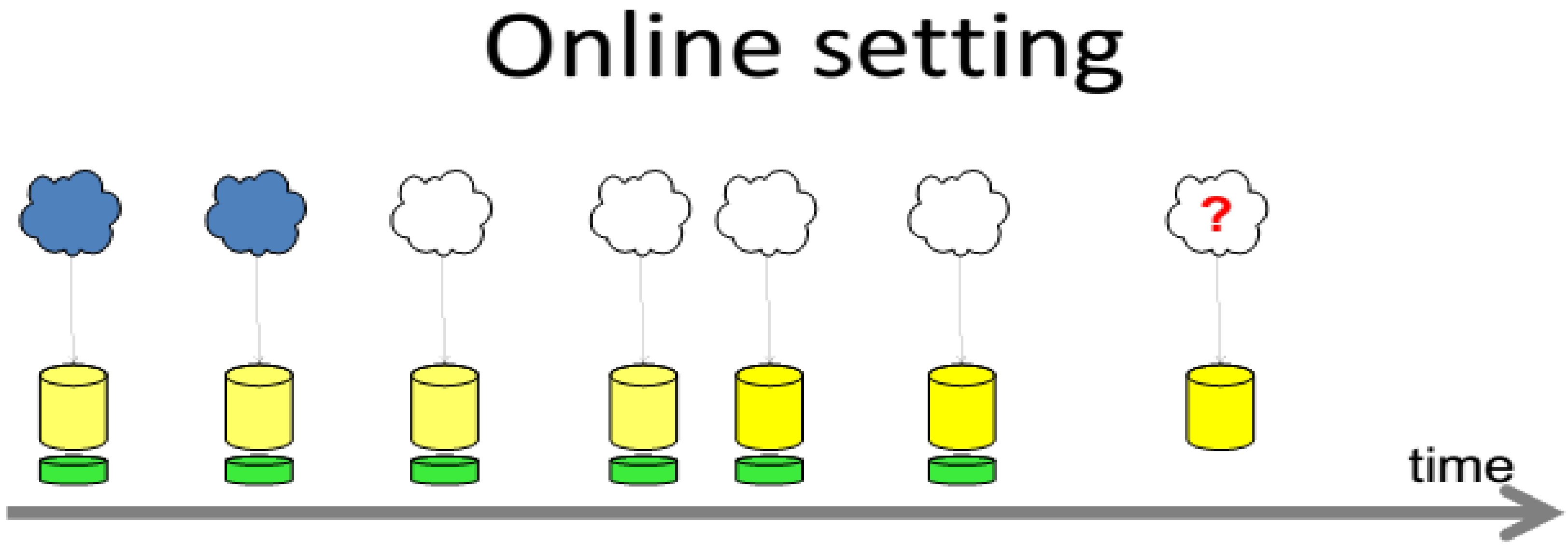
- An alternative to retraining an entire model is to develop a model that adaptively learns from the changing data.
- Such models have the ability to partially update themselves when the underlying data distribution changes [80], as shown in Figure:



A decision tree node is replaced with a new one as its performance deteriorates when a concept drift occurs in a subregion

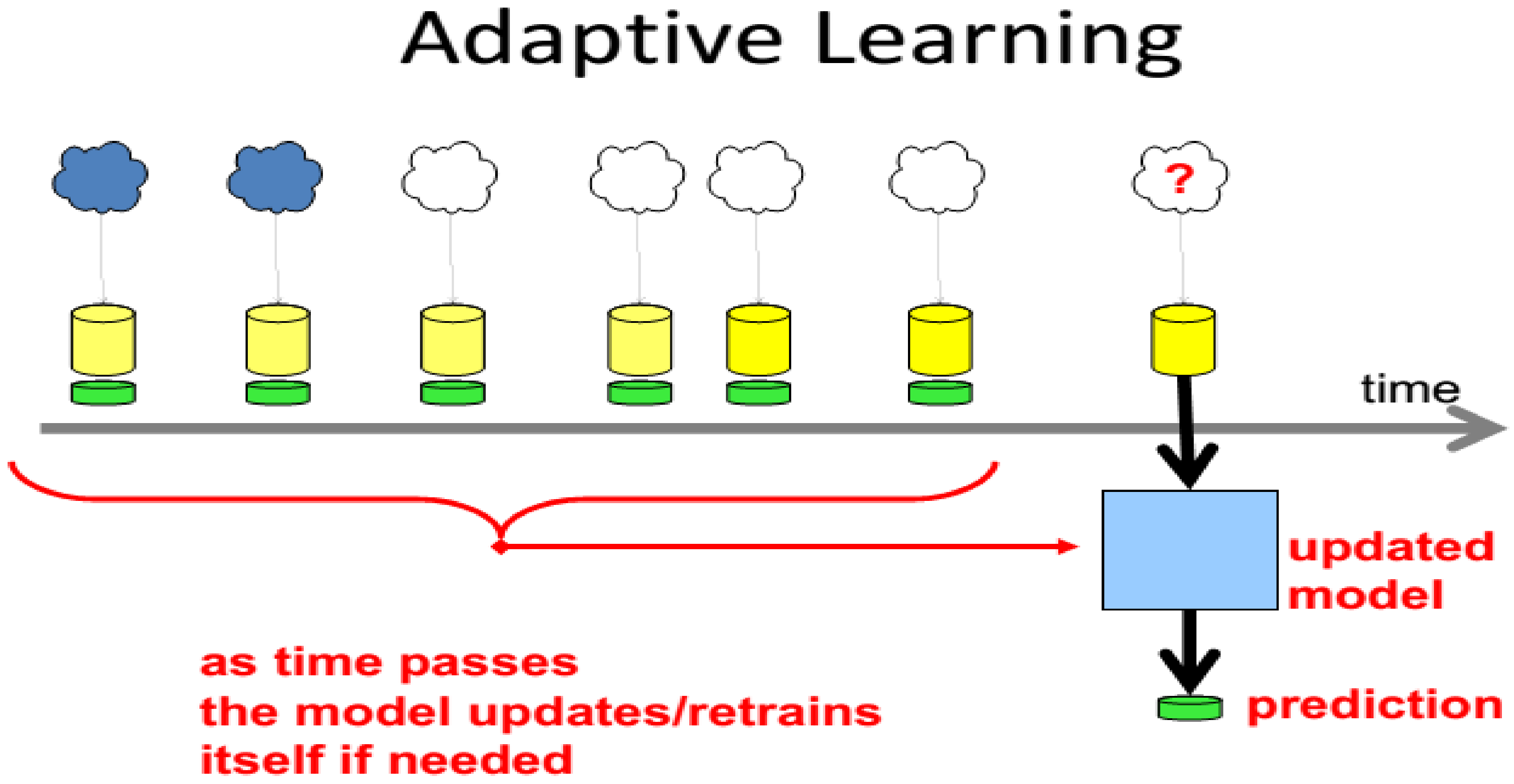
- This approach is arguably more efficient than retraining when the drift only occurs in local regions.
- Many methods in this category are based on the decision tree algorithm because trees have the ability to examine and adapt to each sub-region separately.

The main strategies how to handle concept drift



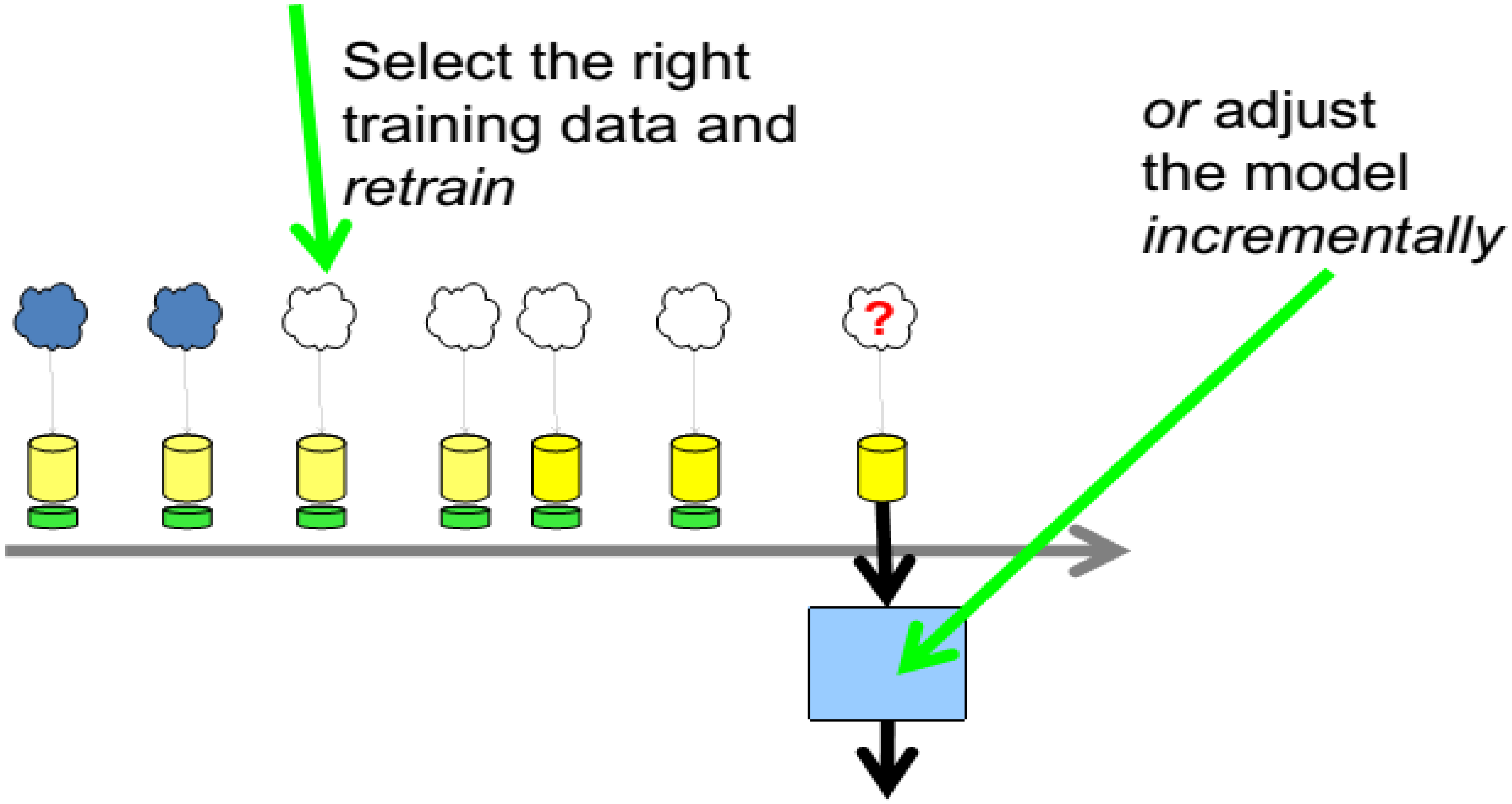
Data arrives in a stream

The main strategies how to handle concept drift



The main strategies how to handle concept drift

How to Adapt

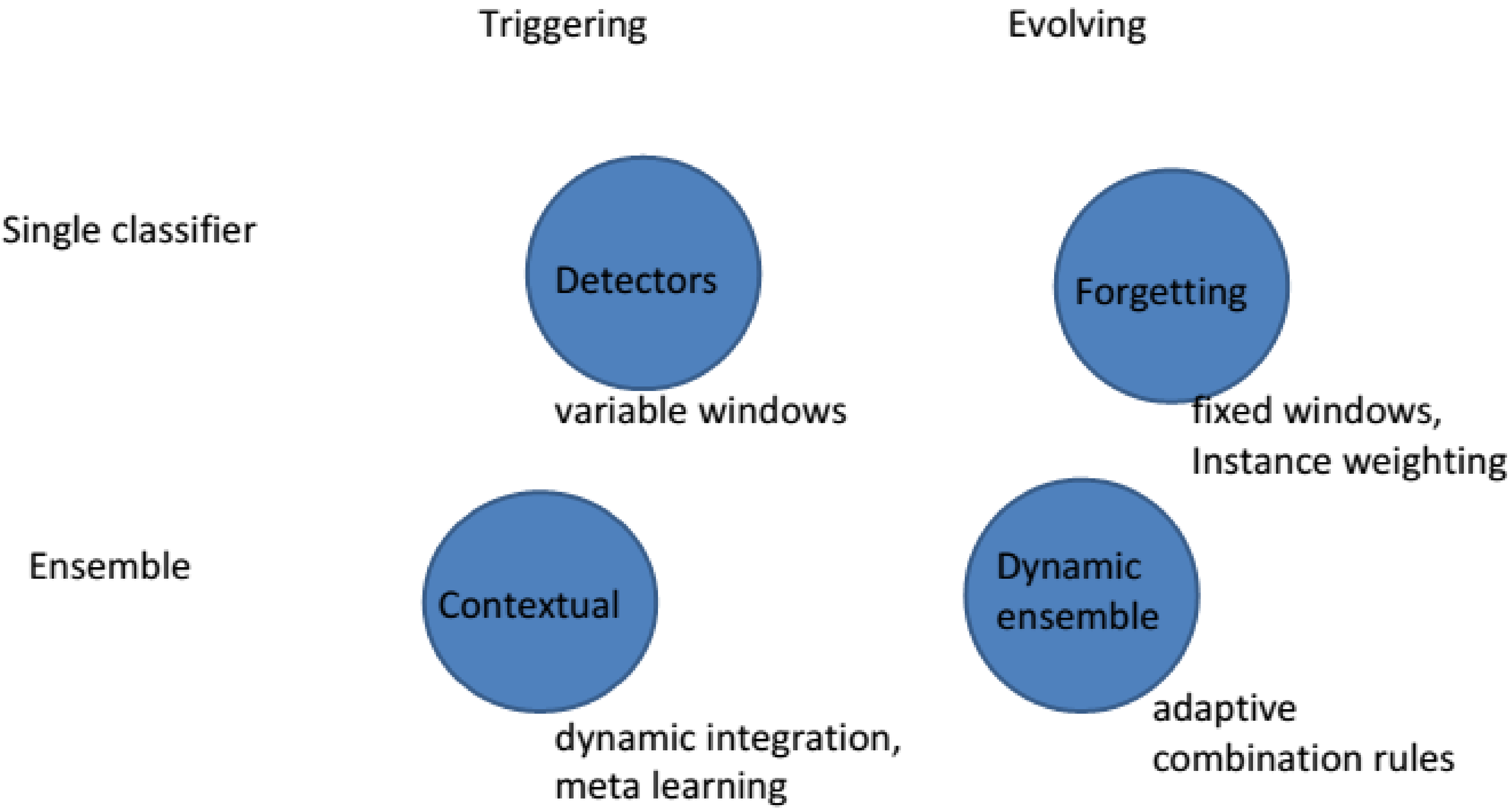


Select the right training data and *retrain*

or adjust the model *incrementally*

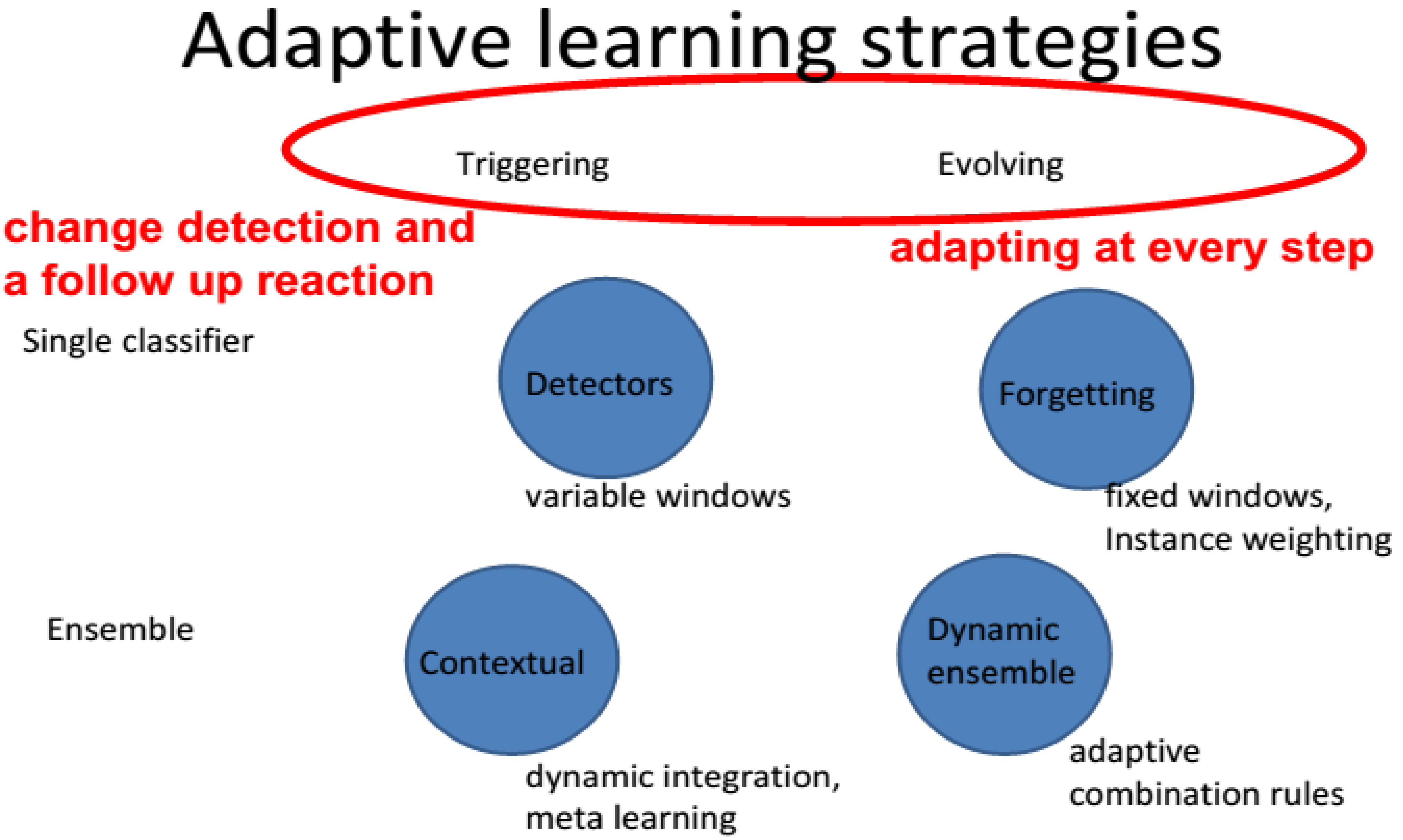
The main strategies how to handle concept drift

- 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

The main strategies how to handle concept drift



Adaptive learning strategies

Triggering

Evolving

change detection and a follow up reaction

adapting at every step

Single classifier

Detectors

variable windows

Forgetting

fixed windows, Instance weighting

Ensemble

Contextual

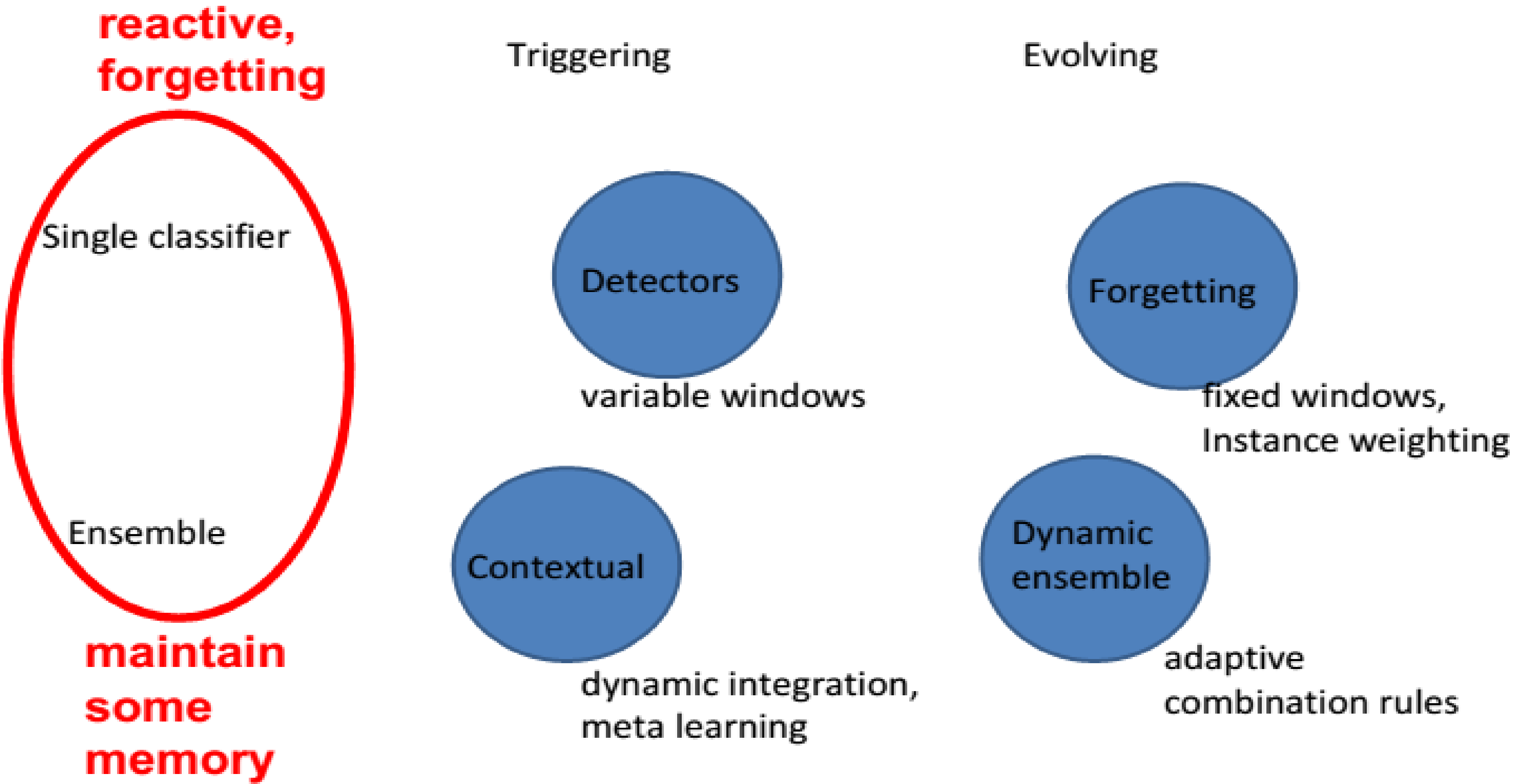
dynamic integration, meta learning

Dynamic ensemble

adaptive combination rules

The main strategies how to handle concept drift

Adaptive learning strategies



reactive, forgetting

Triggering

Evolving

Single classifier

Detectors

variable windows

Forgetting

fixed windows, Instance weighting

Ensemble

Contextual

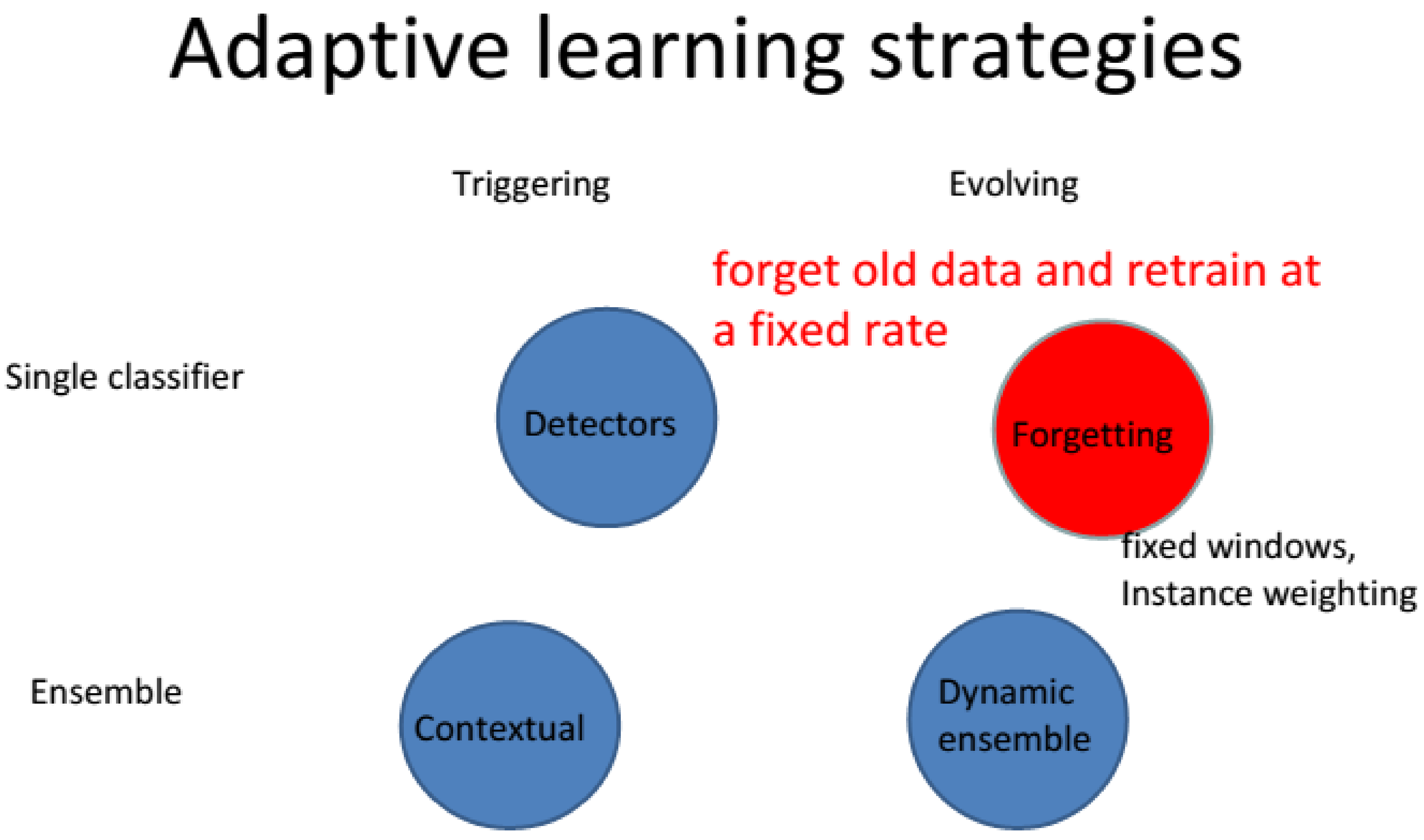
dynamic integration, meta learning

Dynamic ensemble

adaptive combination rules

maintain some memory

The main strategies how to handle concept drift



Adaptive learning strategies

Triggering

Evolving

forget old data and retrain at a fixed rate

Single classifier

Detectors

Forgetting

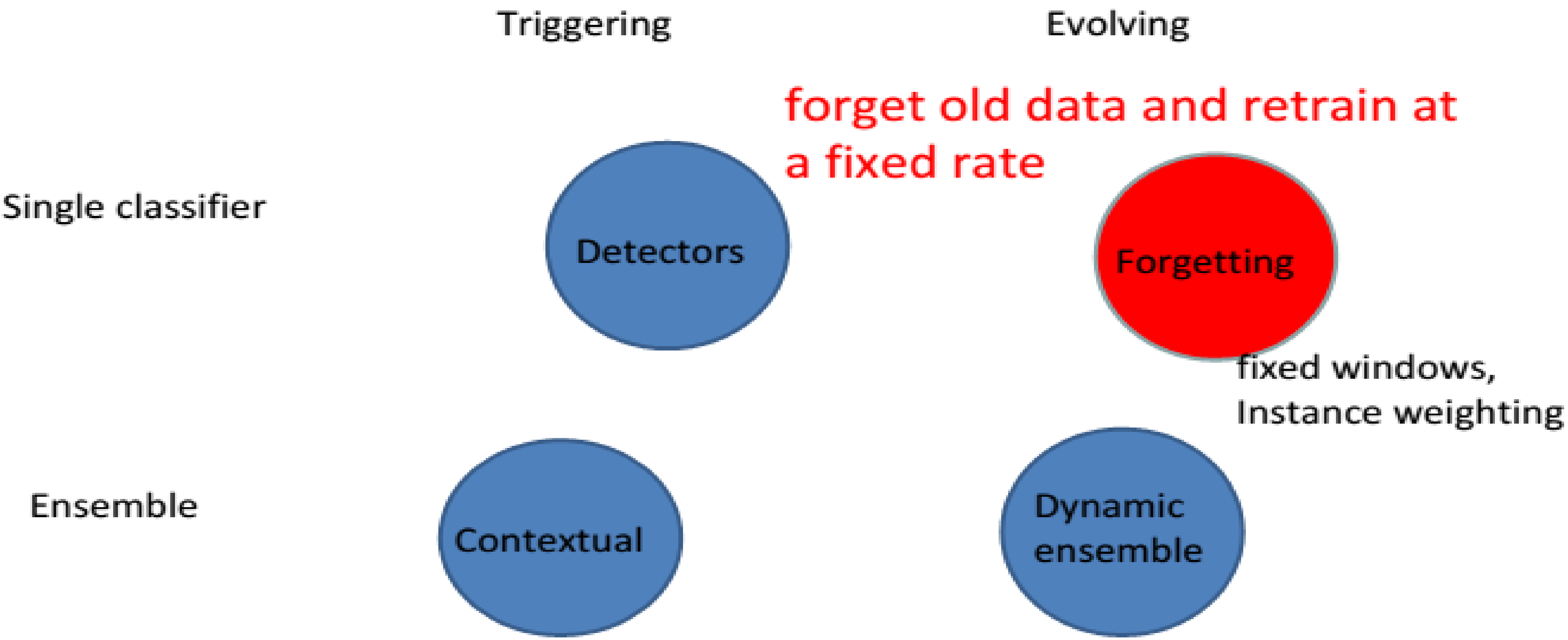
fixed windows,
Instance weighting

Ensemble

Contextual

Dynamic ensemble

Adaptive learning strategies



Triggering

Evolving

forget old data and retrain at a fixed rate

Single classifier

Detectors

Forgetting

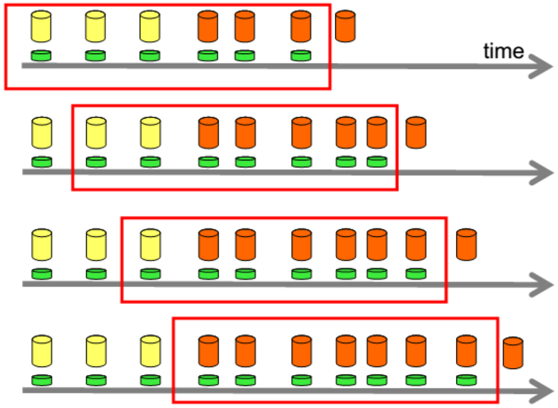
fixed windows,
Instance weighting

Ensemble

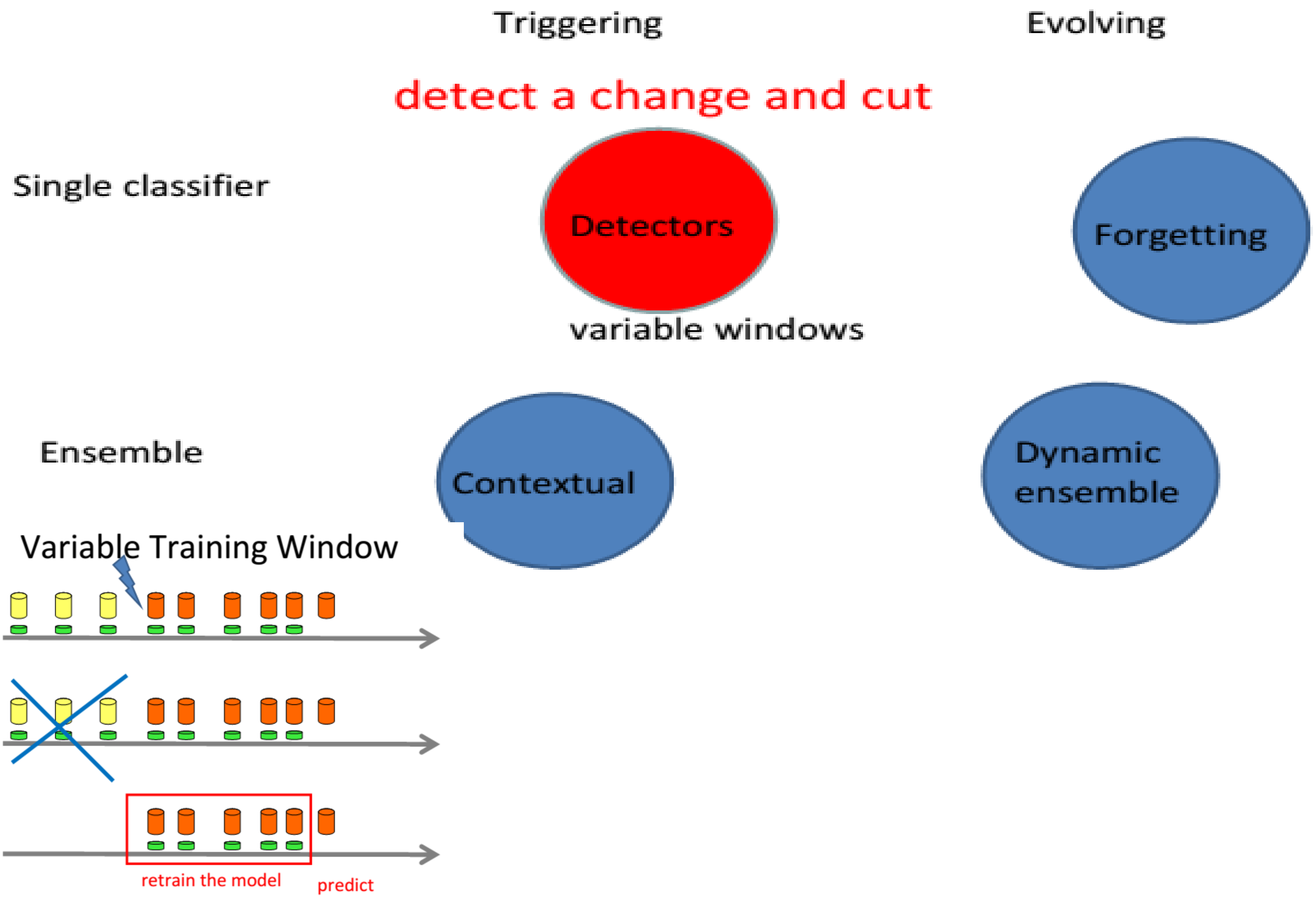
Contextual

Dynamic ensemble

Fixed Training Window



Adaptive learning strategies



Triggering

Evolving

detect a change and cut

Single classifier

Detectors

Forgetting

variable windows

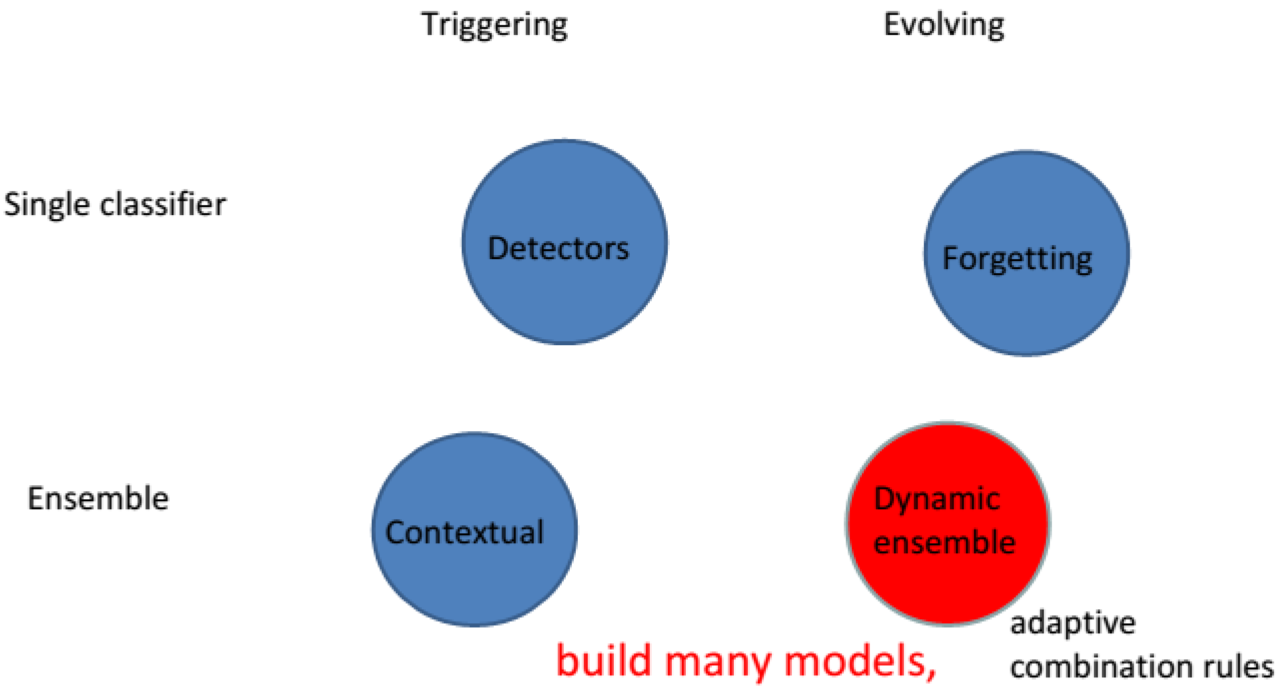
Ensemble

Contextual

Dynamic ensemble

Variable Training Window

Adaptive learning strategies



Triggering

Evolving

Single classifier

Detectors

Forgetting

Ensemble

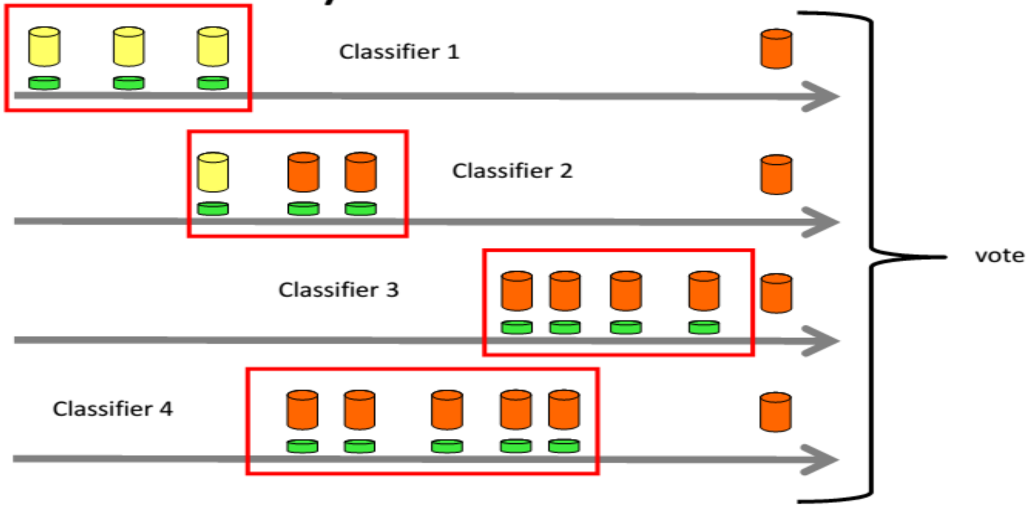
Contextual

Dynamic ensemble

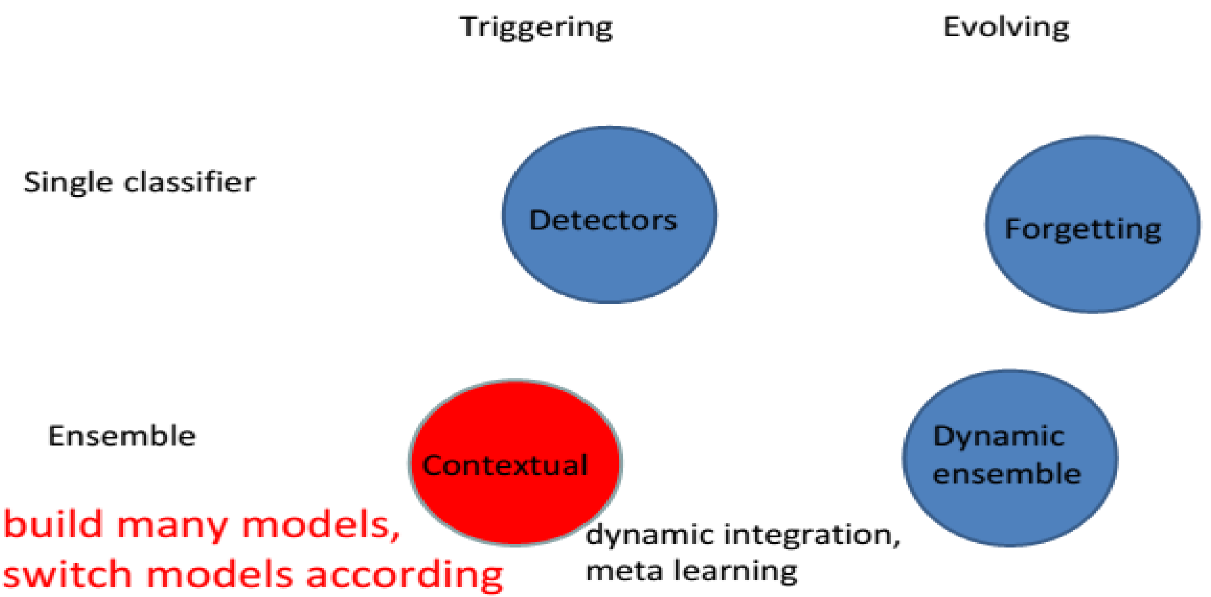
adaptive combination rules

build many models, dynamically combine

Dynamic Ensemble



Adaptive learning strategies


Triggering

Evolving

Single classifier

Detectors

Forgetting

Ensemble

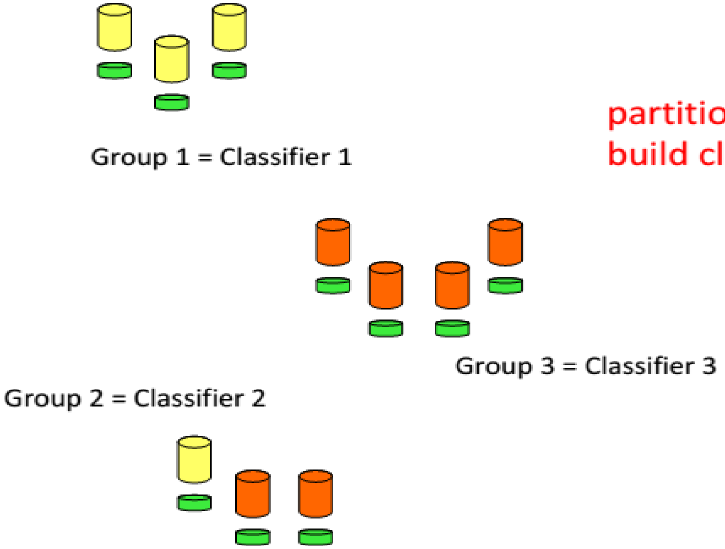
Contextual

Dynamic ensemble

dynamic integration, meta learning

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

Contextual (Meta) Approaches

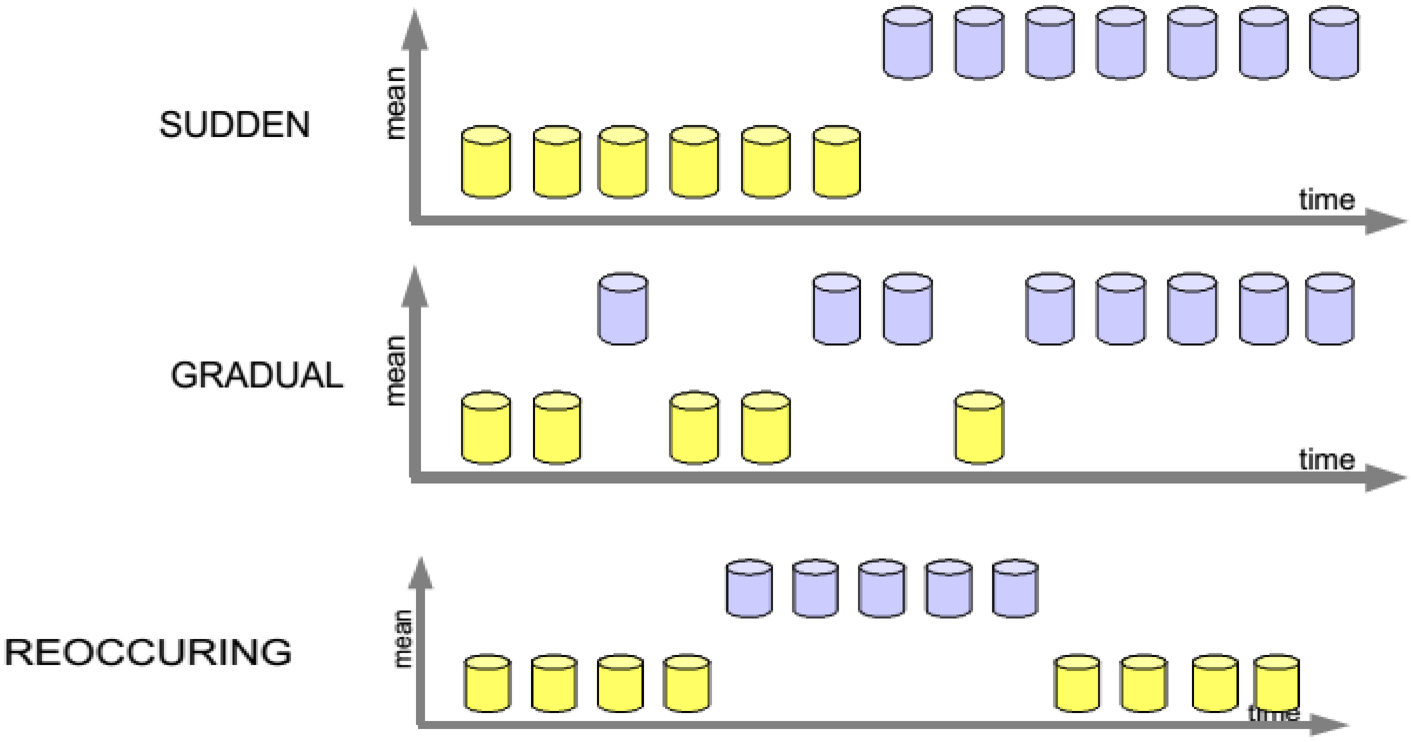


partition the training data
build classifiers

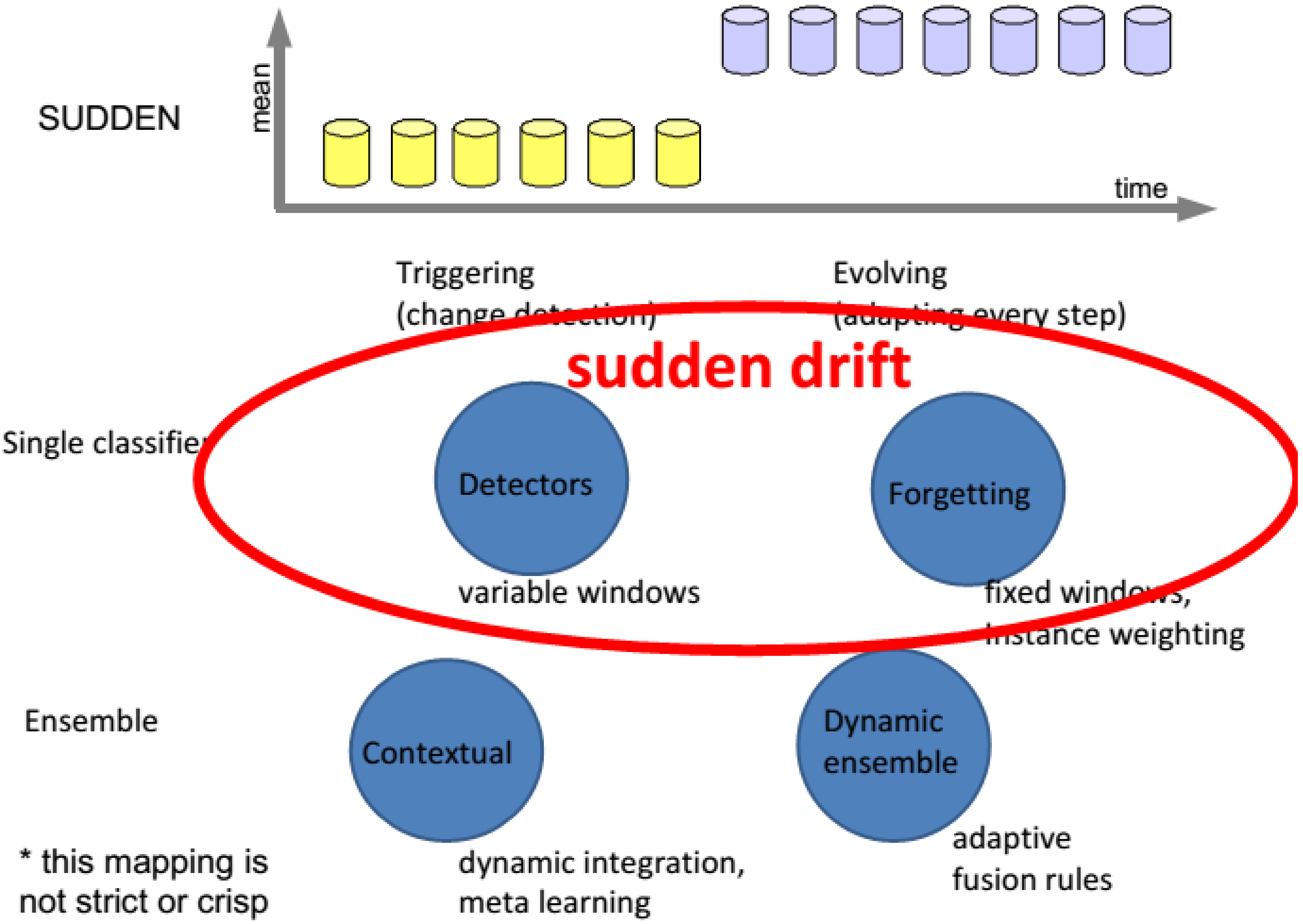
Types of Concept Drifts

- adaptive learning approaches implicitly or explicitly assume some type of change

Types of Changes: speed



Types of Concept Drifts



SUDDEN

Triggering
(change detection)

Evolving
(adapting every step)

sudden drift

Single classifier

Detectors

variable windows

Forgetting

fixed windows,
instance weighting

Ensemble

Contextual

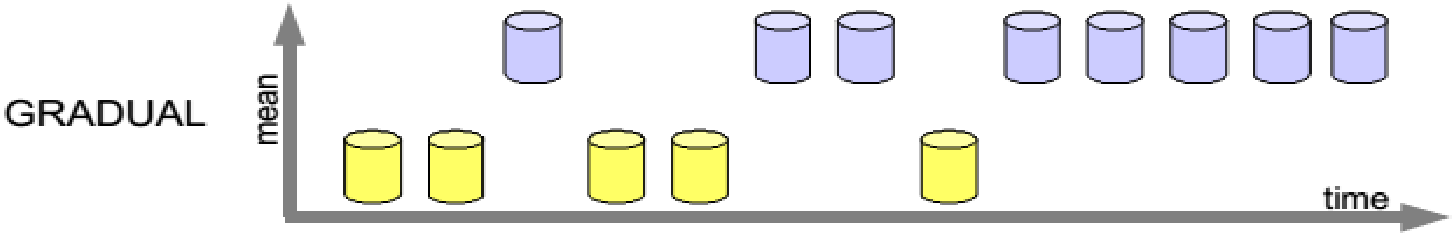
dynamic integration,
meta learning

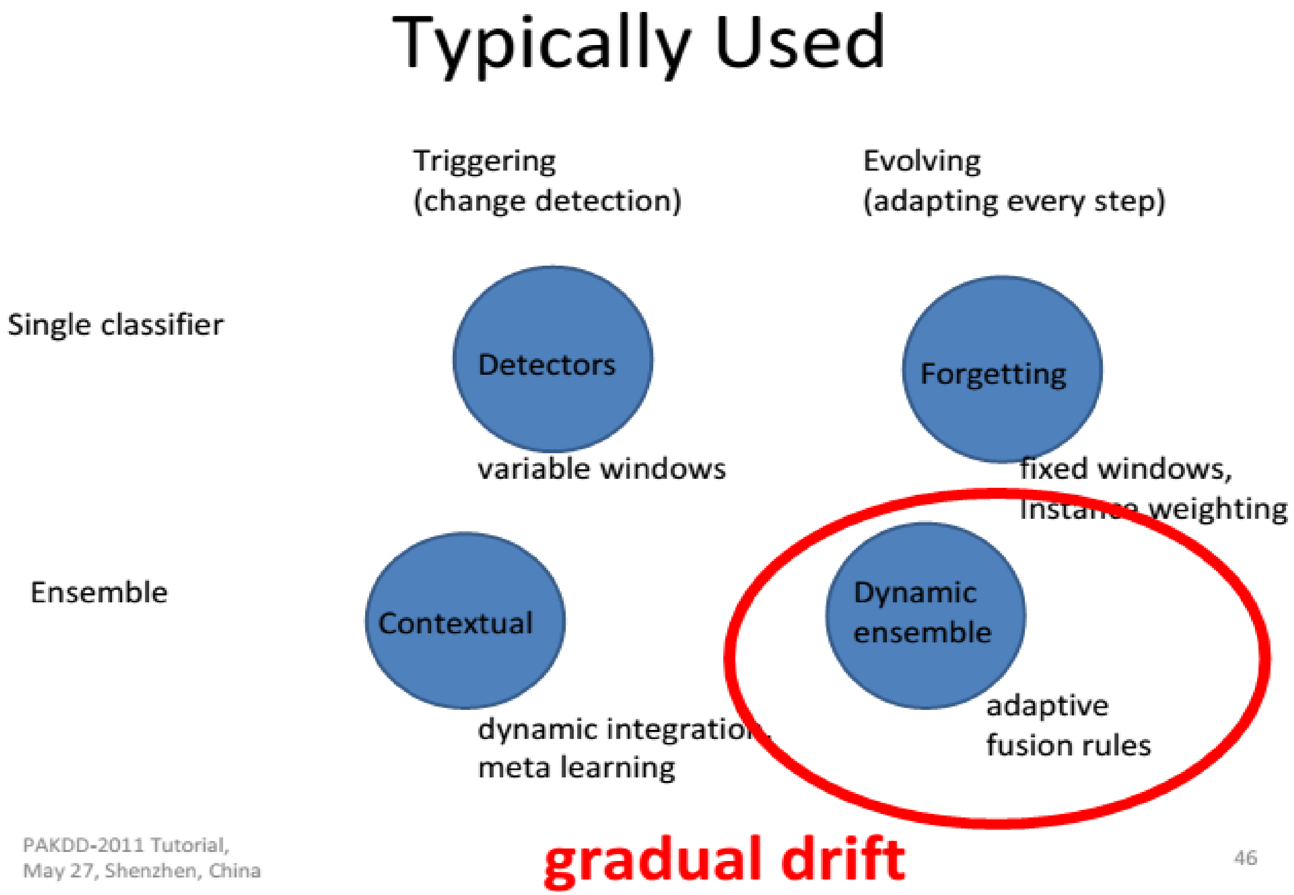
Dynamic ensemble

adaptive
fusion rules

* this mapping is
not strict or crisp

Types of Concept Drifts





Typically Used

	Triggering (change detection)	Evolving (adapting every step)
Single classifier	Detectors variable windows	Forgetting fixed windows, Instance weighting
Ensemble	Contextual dynamic integration, meta learning	Dynamic ensemble adaptive fusion rules

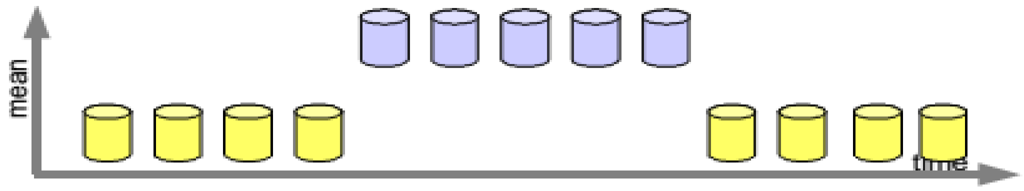
PAKDD-2011 Tutorial,
May 27, Shenzhen, China

46

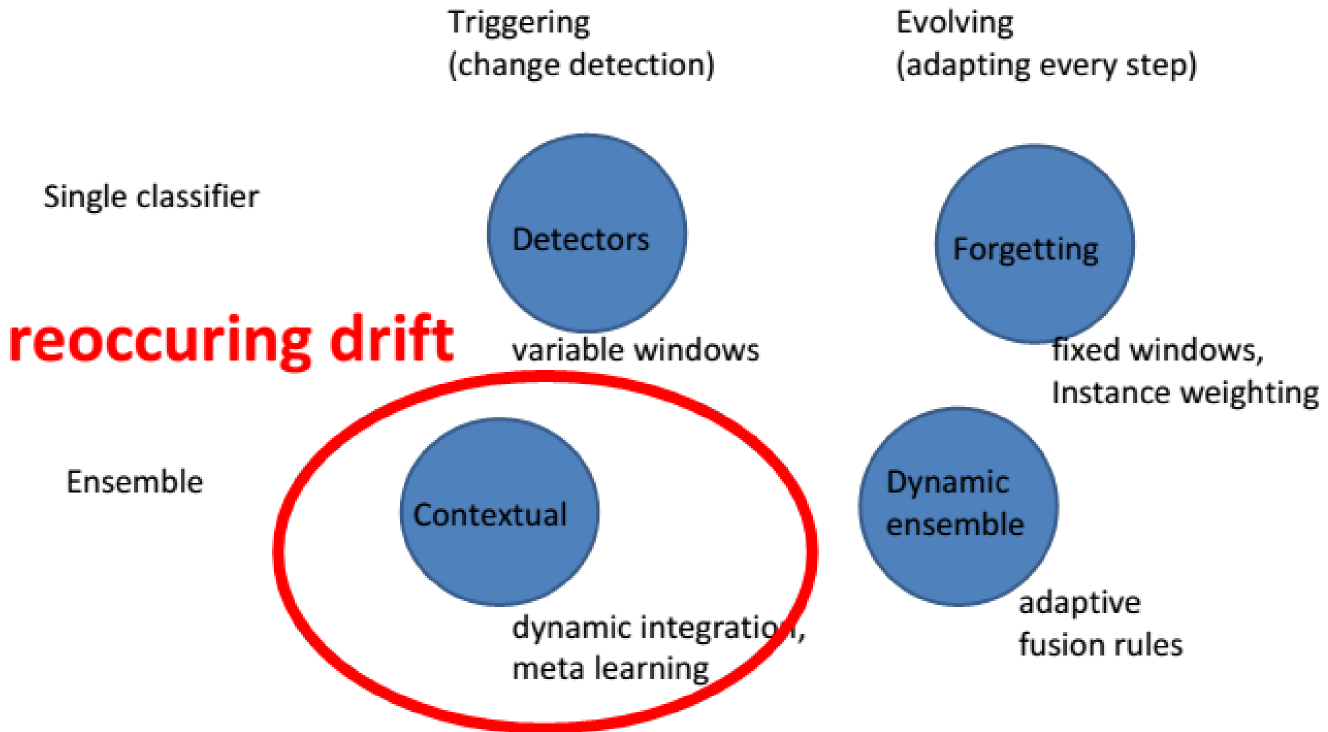
gradual drift

Types of Concept Drifts

REOCCURING



Typically Used



Triggering
(change detection)

Evolving
(adapting every step)

Single classifier

Detectors

variable windows

Forgetting

fixed windows,
Instance weighting

reoccuring drift

Ensemble

Contextual

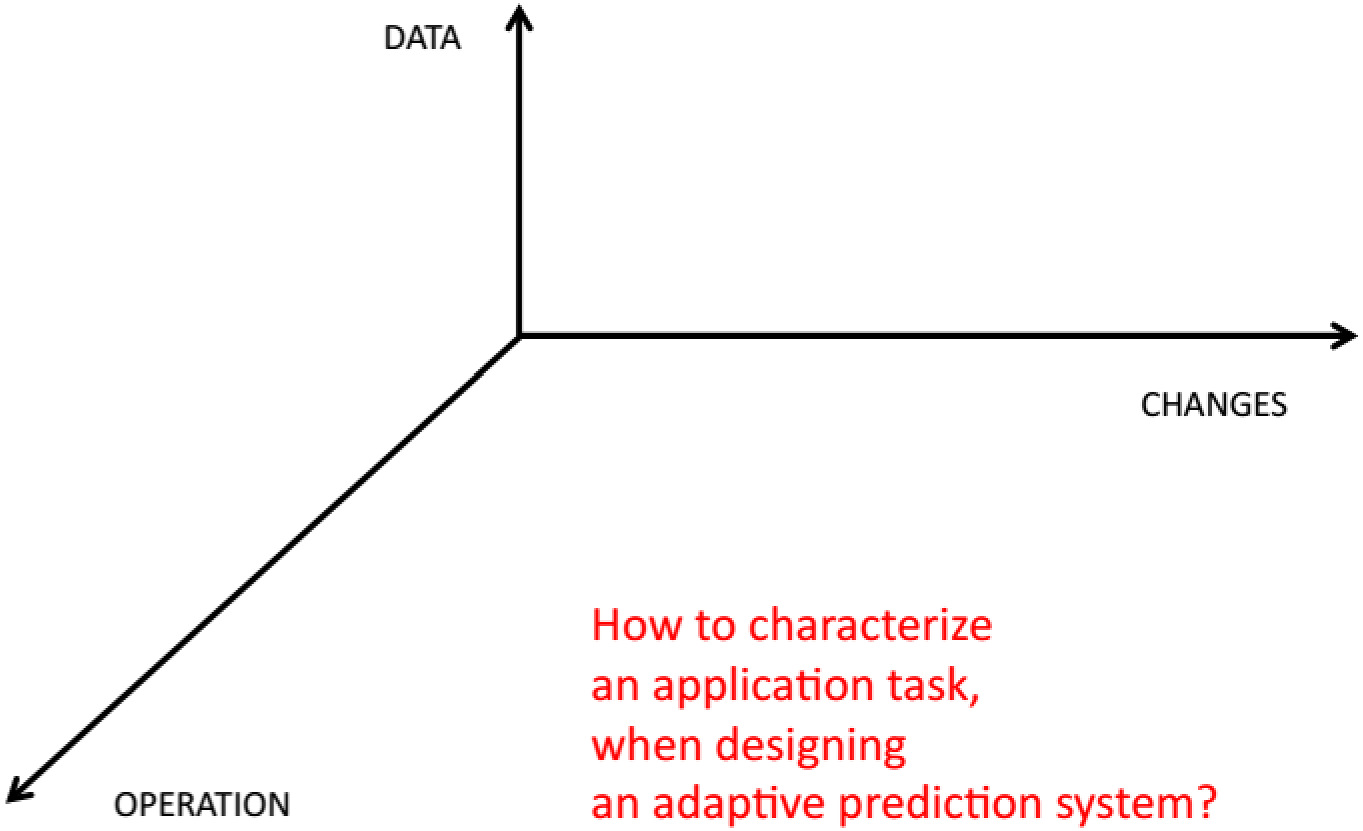
dynamic integration,
meta learning

Dynamic ensemble

adaptive
fusion rules

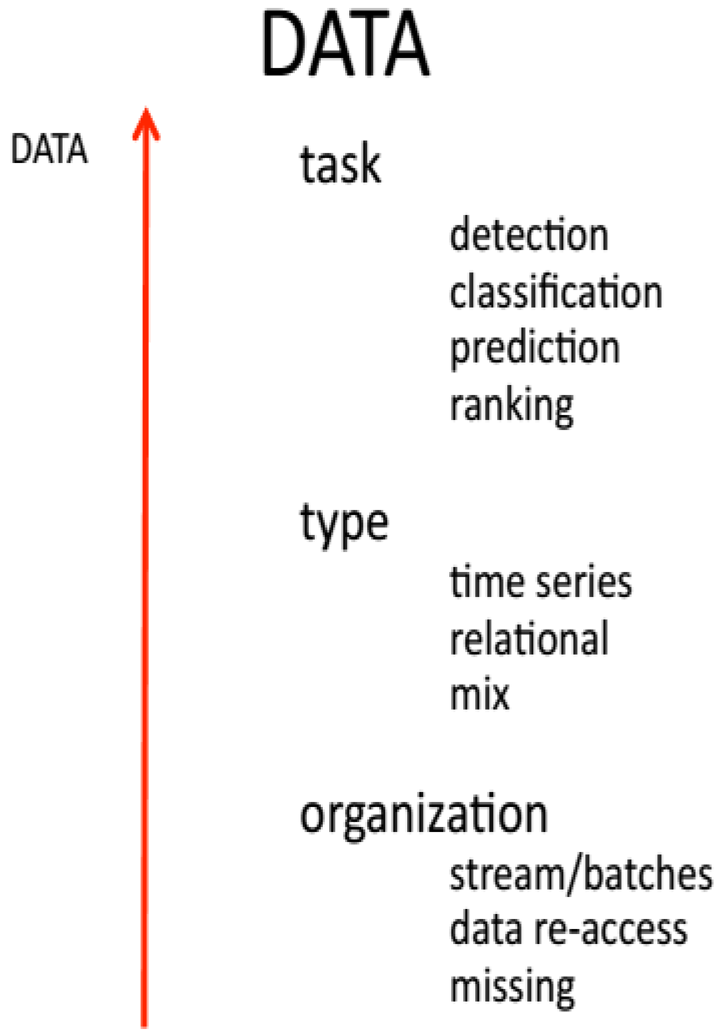
Challenges due to concept drift

Application tasks



How to characterize
an application task,
when designing
an adaptive prediction system?

Challenges due to concept drift



DATA

task
- detection
- classification
- prediction
- ranking

type
- time series
- relational
- mix

organization
- stream/batches
- data re-access
- missing

Challenges due to concept drift

CHANGE

DATA
CHANGES

change type

- sudden
- gradual
- incremental
- reoccurring

change source

- adversary
- interests
- population
- complexity

expectations about changes

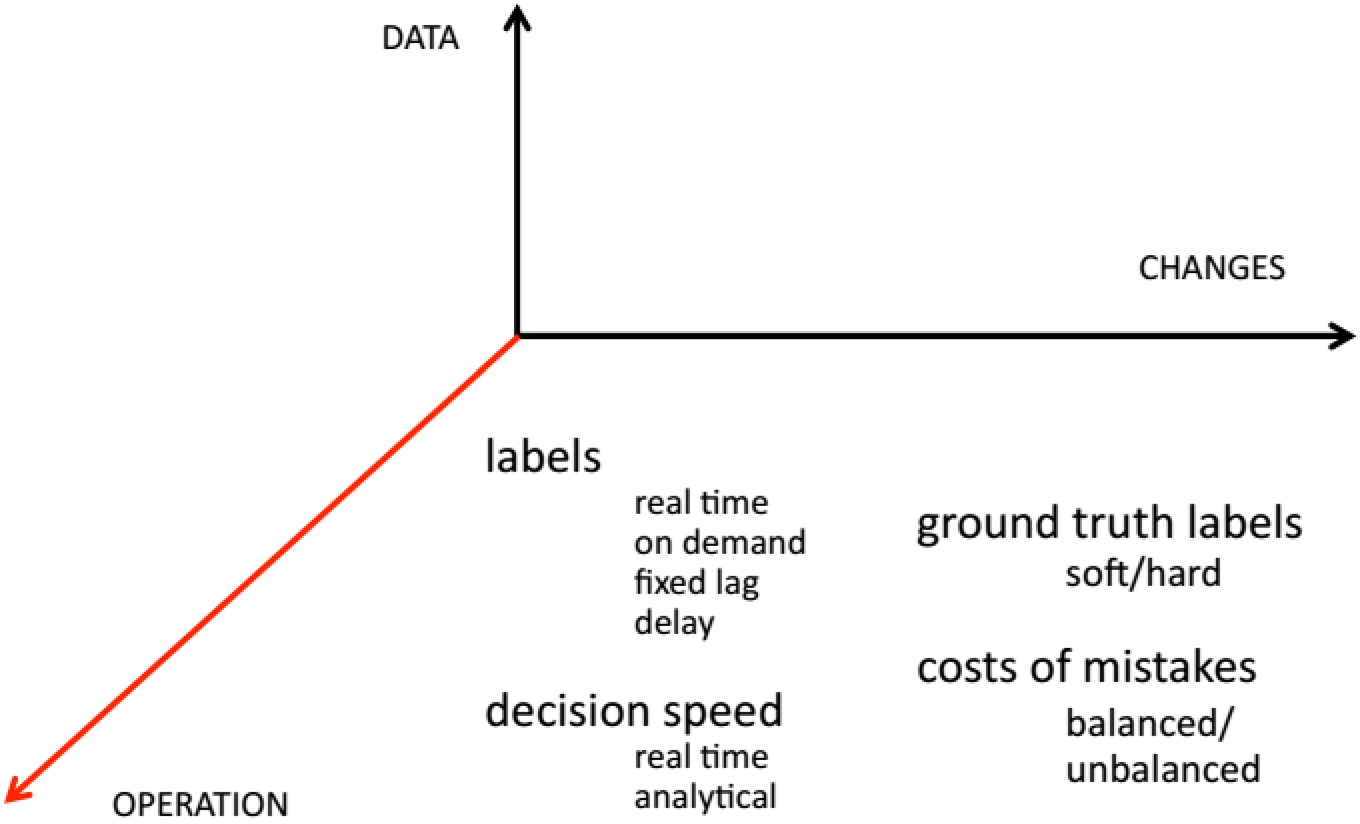
- unpredictable
- predictable
- identifiable

expectations about desired action

- keep the model uptodate
- detect the change
- identify/locate the change
- explain the change

Challenges due to concept drift

OPERATION



labels

- real time
- on demand
- fixed lag
- delay

decision speed

- real time
- analytical

ground truth labels

- soft/hard

costs of mistakes

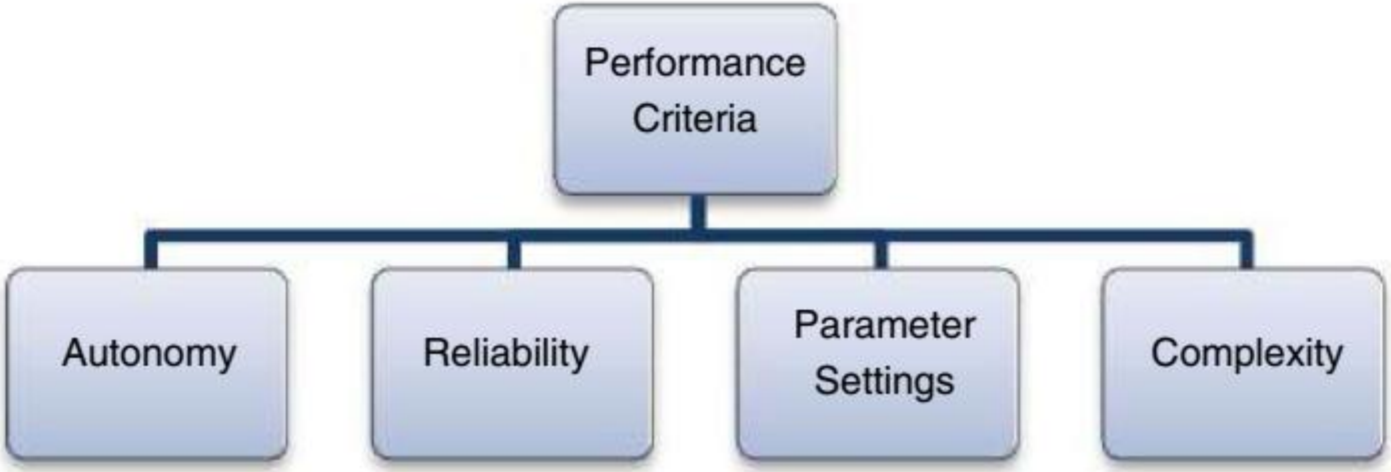
- balanced/ unbalanced

Evaluation Systems

- Several criteria:
  - Time → seconds
  - Memory → RAM/hour
  - Generalizability of the model → % success
  - Detecting concept drift → detected drifts, false positives and false negatives

Evaluation Systems

- Taxonomy of performance criteria for handling concept drift
- According to the requirements of the real world applications, the performance criteria can concern:
  — Autonomy: the level of human involvement,
  — Reliability: the accuracy of drift information,
  — Parameter settings: the availability of *a priori* knowledge,
  — Complexity: the time and memory consumption



PERFORMANCE EVALUATION PARAMETERS OF STREAM PROCESSING

Kappa statistics

- Kappa statistics measure the performance of streaming classifiers and is effective for measuring performance of **imbalanced data sets** wherein number of data instances from one class beats the number of instances from other classes significantly.

$$k = \frac{A_{ref} - A_{rand}}{1 - A_{rand}}$$

- Here, A_{ref} represents the accuracy of the reference classifier which is being evaluated and A_{rand} is Random classifier's accuracy. Kappa values lies in range [0, 1] or sometimes represented in form of percentage range [0%, 100%]. Higher value implies better performance.

Temporal Kappa statistics

- This statistic measures the effectiveness of classifier in the presence of temporal dependence in the data instances of streaming data wherein the class label of data instance at time t+1 tends to belong to the same class as of data instance at time t. The kappa temporal statistic is defined as:

$$k_{temp} = \frac{A_{ref} - A_{pers}}{1 - A_{pers}}$$

- Here, A_{pers} is Persistent classifier's accuracy which predicts same class label of data instance at time t+1 as of data instance at time t. The value of k_{temp} ranges between interval (1, -∞). The k_{temp} = 1 if the classifier is accurate. Negative values of k_{temp} tell that the performance of the classifier is even worse than the persistent classifier.

Classification Parameters

Evaluation Measure	**Major Purpose**	**Value Significance**
Kappa statistics	Assess performance in imbalanced data stream case	Higher value means better performance
Temporal Kappa statistics	Assess performance in case of temporal dependent data streams	Negative values means worse performance

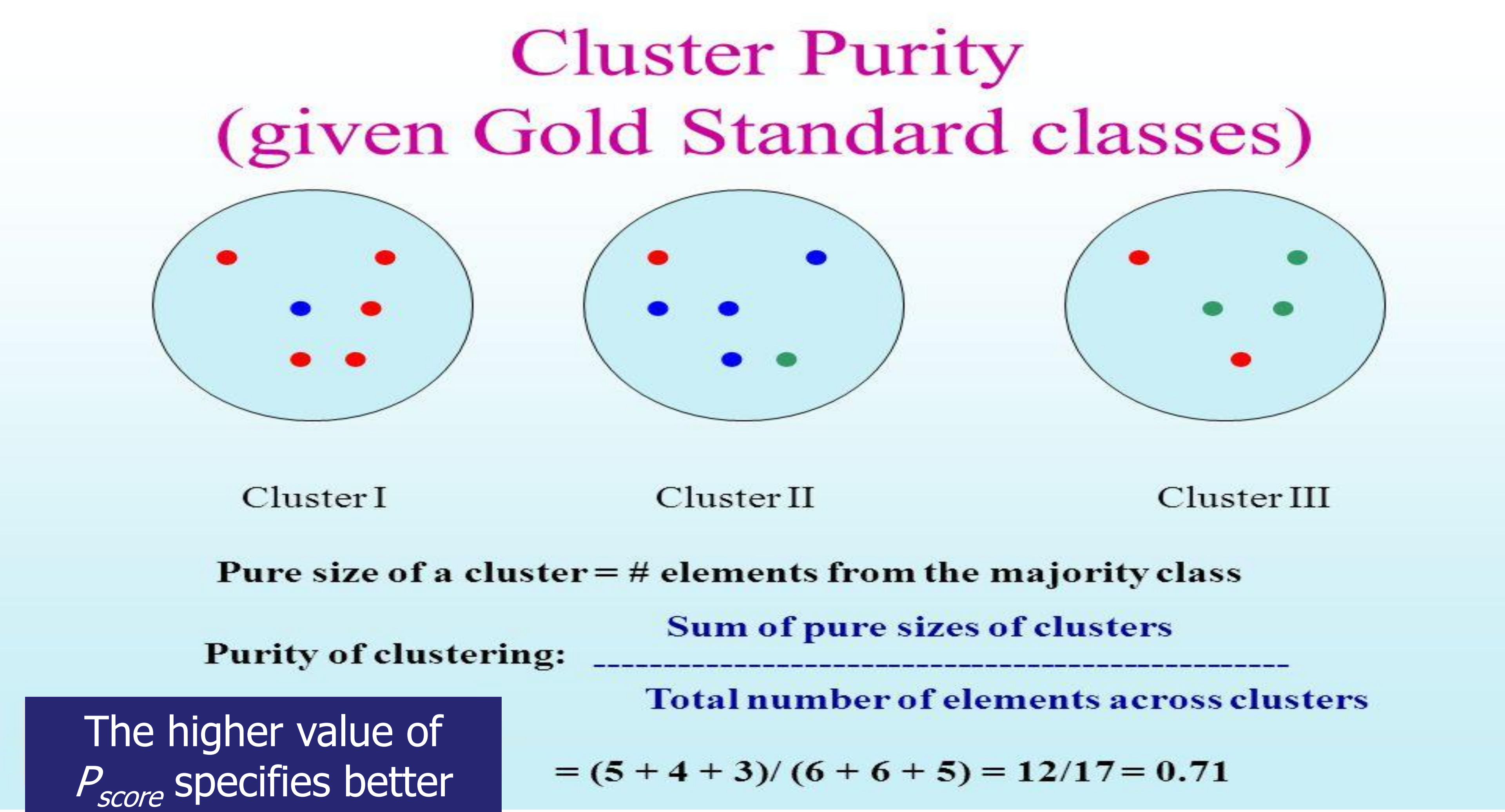
Completeness

- It does assessment that all the data instances belonging to the same class lie in the same cluster or not. For e.g., consider a dataset D composed of data instances belonging to single category. Let one clustering algorithm A_1 generates two clusters C_1 and C_2 whereas another clustering algorithm A_2 produces a single cluster C. Then it can be represented as per equation:

- ***Completeness (cluster-set $\{C\}$) > Completeness (clusters-set $\{C_1, C_2\}$)***

- Values for completeness parameter lies in [0, 1], where higher values implies better performance.

Purity



Cluster Purity (given Gold Standard classes)

Pure size of a cluster = # elements from the majority class

Purity of clustering: $$\frac{\textbf{Sum of pure sizes of clusters}}{\textbf{Total number of elements across clusters}}$$

$$= (5 + 4 + 3)/ (6 + 6 + 5) = 12/17= 0.71$$

The higher value of P_{score} specifies better performance.

SSQ

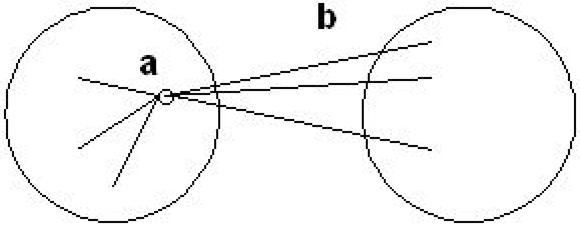
- *SSQ:* It measures cohesiveness of the clusters by computing the sum of the square of distance of each instance in the cluster from their respective centroid (Song and Zhang, 2013). It is calculated for each cluster as indicated in equation .

$$SSQ = \sum_{j} \sum_{i=1}^{n} d_{i,c_j}^2$$

- Here, n specifies the number of data instances in cluster j and $d_{i,cj}$ is the distance of instance i from cluster centrec_j of the j^{th} cluster.
- The smaller value of SSQ implies better performance.

Silhouette Coefficient

- **Silhouette Coefficient combine ideas of both cohesion and separation, but for individual points**
  - Calculate ***a*** = average distance of *i* to the points in its cluster
  - Calculate ***b*** = min (average distance of *i* to points in another cluster)
  - The silhouette coefficient for a point is s = 1 – a/b if a < b, (or s = b/a - 1 if a ≥ b, not the usual case)
  - Typically between 0 and 1
  - The closer to 1 the better



- **Can calculate the average Silhouette width for a cluster or a clustering**

Clustering Parameters

Completeness	Measures whether same class instance fall in same cluster or not	Higher value means better clustering
Purity	Assesses purity of the clusters in terms of having same class instances	Higher value means better clustering
SSQ	Measures clusters cohesiveness	Lower value means better clustering
Silhouette Coefficient	Assess compactness as well as separation of clusters	Higher value means better clustering

Important findings

- Error rate-based and data distribution-based drift detection methods are still playing a dominant role in concept drift detection research, while multiple hypothesis test methods emerge in recent years;
- Regarding to concept drift understanding, all drift detection methods can answer “When”, but very few methods have the ability to answer “How” and “Where”;

Important findings

- Adaptive models and ensemble techniques have played an increasingly important role in recent concept drift adaptation developments. I
- n contrast, research of retraining models with explicit drift detection has slowed;
- Most existing drift detection and adaptation algorithms assume the ground true label is available after classification/prediction, or extreme verification latency.
- Very few research has been conducted to address unsupervised or semi-supervised drift detection and adaptation

important findings

- Some computational intelligence techniques, such as fuzzy logic, competence model, have been applied in concept drift.
- There is no comprehensive analysis on real-world data streams from the concept drift aspect, such as the drift occurrence time, the severity of drift, and the drift regions.

RECENT TRENDS AND FUTURE PERSPECTIVE

From Algorithms Development Point of View

- The development of new algorithms that addresses the inherent challenges in mining large scale data streams. New algorithms must ensure:
  - One-pass computation over the stream of data.
  - Faster computation to respond in real time.
  - Minimizing the memory utilization by storing the summarized or sampled data information without significantly losing the accuracy of mining result.

From New Evaluation Measures Point of View

- Traditional evaluation measures are not sufficient to estimate the performance of the stream mining tasks. Hence, identification of new evaluation measures is also an important field of research in stream data mining. These measures must consider:
  - Underlying imbalances in data sets
  - Non-uniform distribution of incoming data instances.
  - Temporal dependence of data instances.

From Concept Change Identification Point of View

- In streaming data mining, the change of concept is the common phenomenon.
- It opens a plenty of opportunities for research. Mining techniques must be capable of identifying these concept changes with time.
- Also, mining techniques must periodically update the model or take the appropriate steps accordingly to capture concept drift and to deal with it.

Future Directions

- Drift detection research should not only focus on identifying drift occurrence time accurately, but also need to provide the information of drift severity and regions. These information could be utilized for better concept drift adaptation.

- In the real-world scenario, the cost to acquire true label could be expensive, that is, unsupervised or semi-supervised drift detection and adaptation could still be promising in the future.

Future Directions

- A framework for selecting real-world data streams should be established for evaluating learning algorithms handling concept drift.
- Research on effectively integrating concept drift handling techniques with machine learning methodologies for data-driven applications is highly desired.

References

- J. Lu, A. Liu, F. Dong, F. Gu, J. Gama and G. Zhang, "Learning under Concept Drift: A Review," in IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 12, pp. 2346-2363, 1 Dec. 2019.
- C. Aggarwal, J. Han, J. Wang, P. S. Yu, “A Framework for Clustering Data Streams”, VLDB'03
- C. C. Aggarwal, J. Han, J. Wang and P. S. Yu, “On-Demand Classification of Evolving Data Streams”, KDD'04
- C. Aggarwal, J. Han, J. Wang, and P. S. Yu, “A Framework for Projected Clustering of High Dimensional Data Streams”, VLDB'04
- S. Babu and J. Widom, “Continuous Queries over Data Streams”, SIGMOD Record, Sept. 2001
- B. Babcock, S. Babu, M. Datar, R. Motwani and J. Widom, “Models and Issues in Data Stream Systems”, PODS'02.
- Y. Chen, G. Dong, J. Han, B. W. Wah, and J. Wang, “Multi-Dimensional Regression Analysis of Time-Series Data Streams”, VLDB'02
- P. Domingos and G. Hulten, “Mining high-speed data streams”, KDD'00
- A. Dobra, M. N. Garofalakis, J. Gehrke, and R. Rastogi, “Processing Complex Aggregate Queries over Data Streams”, SIGMOD’02

References

- S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan, “Clustering Data Streams”, FOCS'00
- G. Hulten, L. Spencer and P. Domingos, “Mining time-changing data streams”, KDD’01
- S. Madden, M. Shah, J. Hellerstein, V. Raman, “Continuously Adaptive Continuous Queries over Streams”, SIGMOD’02
- G. Manku, R. Motwani, “Approximate Frequency Counts over Data Streams”, VLDB’02
- A. Metwally, D. Agrawal, and A. El Abbadi. “Efficient Computation of Frequent and Top-k Elements in Data Streams”. ICDT'05
- S. Muthukrishnan, “Data streams: algorithms and applications”, Proc 2003 ACM-SIAM Symp. Discrete Algorithms, 2003
- R. Motwani and P. Raghavan, ***Randomized Algorithms***, Cambridge Univ. Press, 1995
- S. Viglas and J. Naughton, “Rate-Based Query Optimization for Streaming Information Sources”, SIGMOD’02
- Y. Zhu and D. Shasha. “StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time”, VLDB’02
- H. Wang, W. Fan, P. S. Yu, and J. Han, “Mining Concept-Drifting Data Streams using Ensemble Classifiers”, KDD'03

Some of the good web resources

- Indyk, ’Streaming etc" lecture notes: http://people.csail.mit.edu/indy...
- Feldman et al., On the Complexity of Processing Massive, Unordered, Distributed Data: http://arxiv.org/abs/cs/0611108
- Feldman et al, On Distributing Symmetric Streaming Computations: http://www.google.com/research/p...
- Sarma et al., Estimating PageRank on graph streams: http://portal.acm.org/citation.c...
- Zhang, A Survey on Streaming Algorithms for Massive Graphs:http://www.springerlink.com/cont...
- Vassilvitskii, "Dealing with Massive Data" lecture notes: http://www.cs.columbia.edu/~coms...
- McGregor's publications: Http://www.cs.umass.edu/~mcgregor
- Muthukrishnan, Data Streams: Algorithms and Applications: http://www.cs.rutgers.edu/~muthu/

Thank You