Eötvös Loránd University (ELTE)
Faculty of Informatics (IK)
Pázmány Péter sétány 1/c
1117 Budapest, Hungary

CHANGE DETECTION

Stream mining (SM)

Imre Lendák, PhD, Associate Professor

Chandresh Maurya, PhD, Assistant Professor

2020
Budapest, Hungary

SM course project clarifications

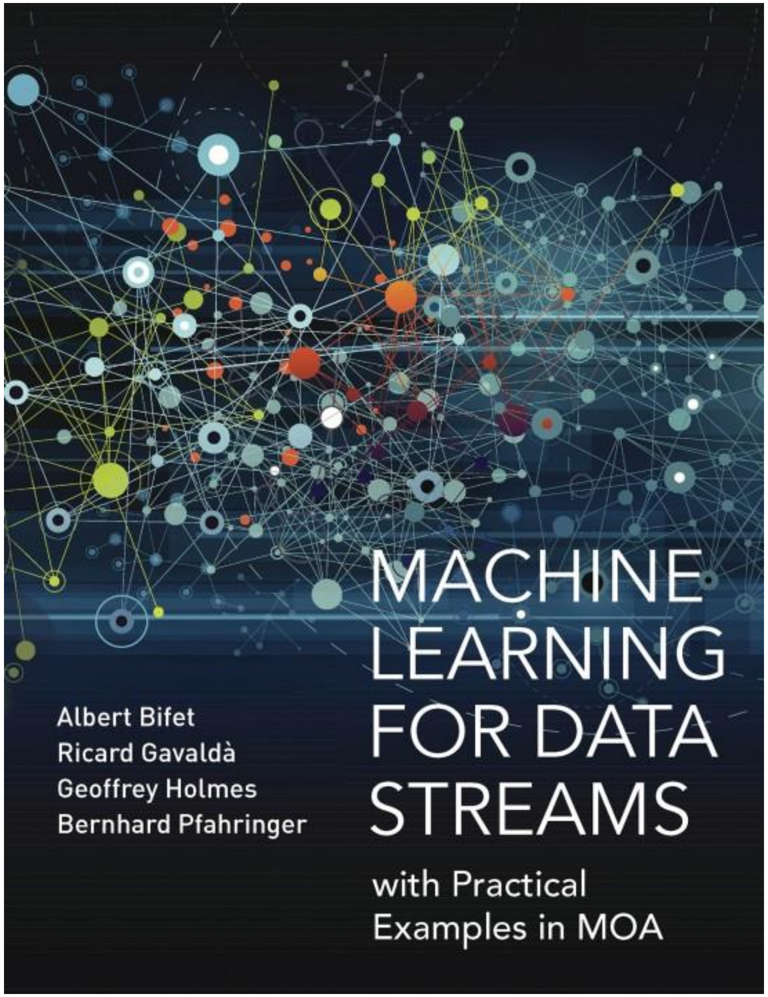
- Each project team member is required to **actively participate** in the streaming system element of the project
  - The separation of duties within the stream mining portion of project should also be completely clear
- If only **simple operations** (e.g. counting and filtering) are used, then the SM project grade will be lower or the lowest
  - The lectures held by Peter and Imre, i.e. everything after Part I: Streaming Systems contains ideas for possible stream processing stages, e.g. sketches, frequent pattern mining, anomaly detection

- **Note:** the use of online systems in the **public cloud** and the upload of course project data into those systems is **strictly forbidden**
  - Any indication of uploading the data in the public cloud or making it available via public & direct links will result in banning the student from the oral exam in school year 2020/2021

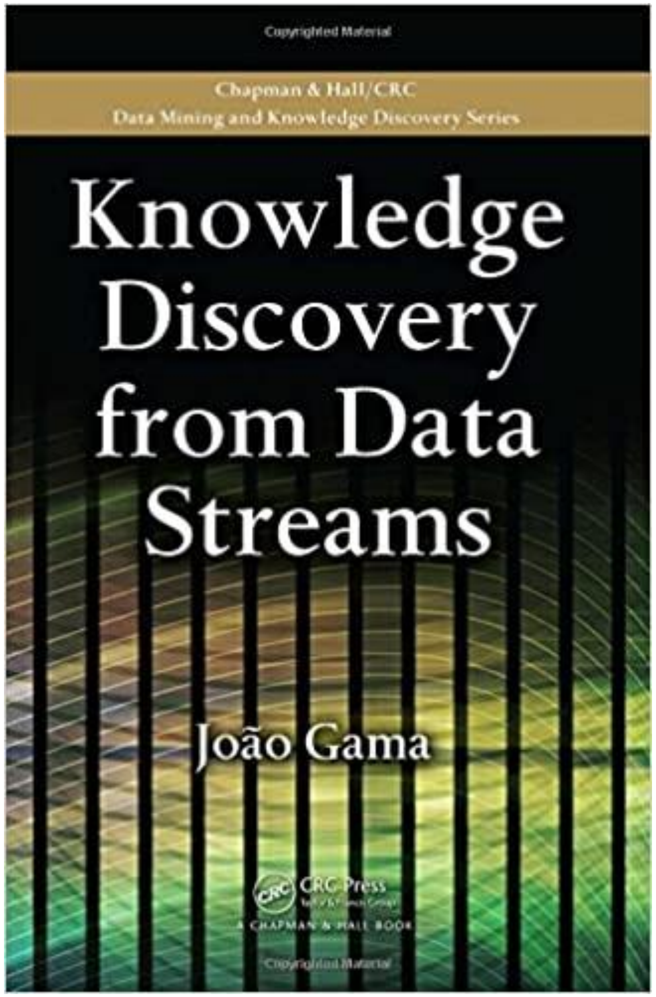
Overview & lecture topics

- Background and terminology
- Responding to change
- Change detection model
  - Detect
  - Adapt
  - Manage models
  - Evaluate
- Use cases
- References

Key sources (but many others as well)



https://mitpress.mit.edu/books/machine-learning-data-streams

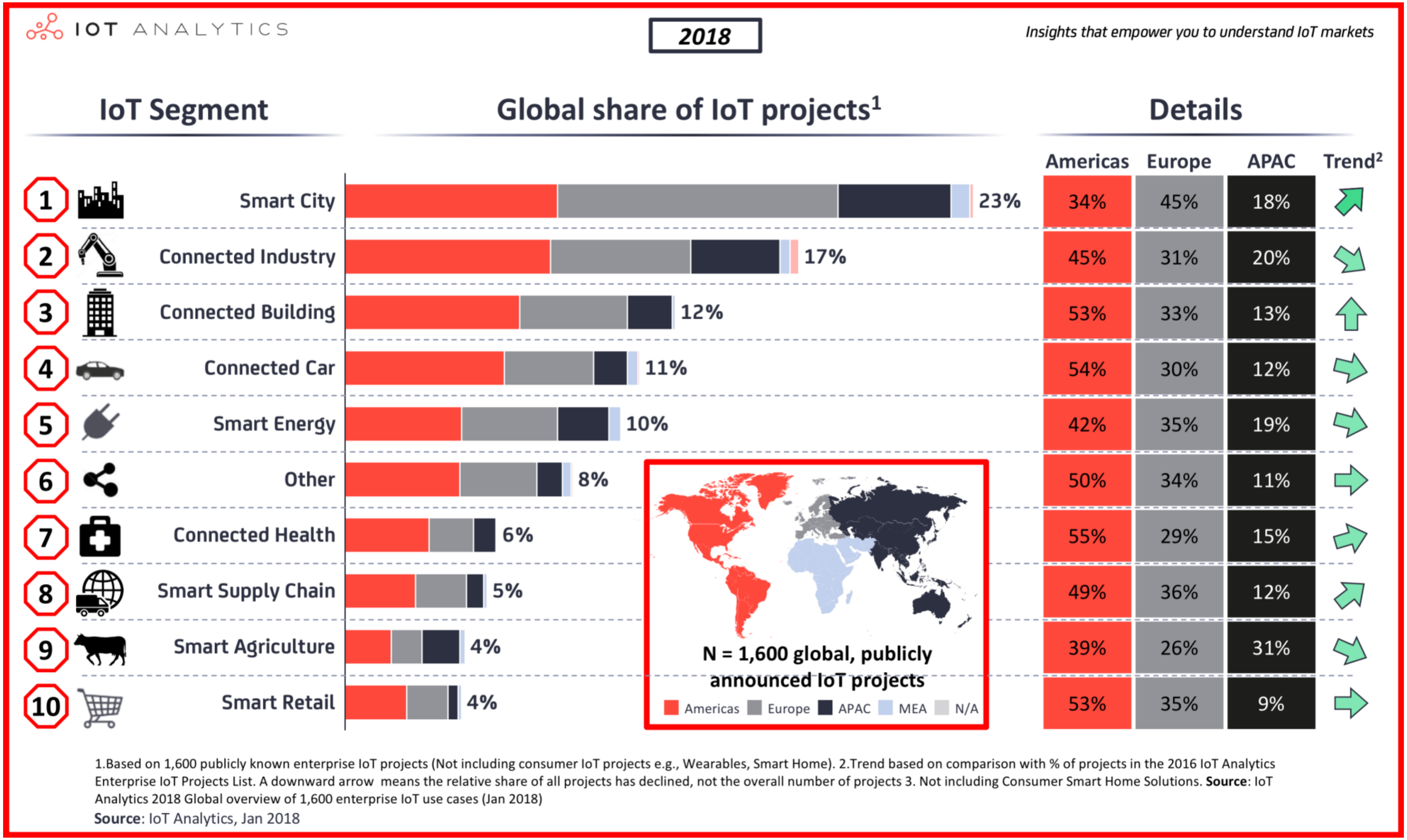


https://www.amazon.com/Knowledge-Discovery-Streams-Chapman-Mining/dp/1439826110

Background

- Historically, most machine learning works assumed that data records used for training are **generated at random** and according to **stationary probability distribution**
- As soon as the 1990s, scientists identified real-life problems in which change detection was highly relevant
  - **User behavior modeling**, e.g. happy employees turn into disgruntled employees
  - **Industrial process monitoring**, e.g. the quality of chemicals used improves over time
  - **Fault detection**, e.g. gradual equipment status deterioration due to material fatigue or other reasons
  - …

Motivation



https://iot-analytics.com/top-10-iot-segments-2018-real-iot-projects/

A world in movement

- The new characteristics of data in the IoT (and other) settings:
  - **Time and space:** the objects of analysis exist in time and space. Often they are able to move.
  - **Dynamic environment:** the objects exist in a dynamic and evolving environment.
  - **Information processing capability:** the objects have limited information processing capabilities.
  - **Locality:** the objects know only their local spatio-temporal environment.
  - **Distributed Environment:** objects will be able to exchange information with other objects.
- Main goal: **real-time analysis** with decision models which **evolve** in correspondence with the evolving environment.

Gama J. Data Sciences: where are we going? Plenary speech @ IDEAL 2020

Challenges of real-time data mining

- Switch from **one-shot learning** to continuously learning dynamic models that evolve over time.
- **Finite training sets, static models, and stationary distributions** will have to be completely thought anew.
- **Computational resources are finite** → algorithms will have to use limited computational resources (in terms of computations, memory, space and time, communications).

- Additional examples:
  - Internet: traffic logs, user queries, email, financial.
  - Telecommunications: phone calls, SMS,
  - Astronomical surveys: optical, radio.
  - Sensor networks: many more observation points.

Gama J. Data Sciences: where are we going? Plenary speech @ IDEAL 2020

From static to real-time

Stream mining yesterday

- In **real-time stream mining** we face continuous data flows generated at high-speed in dynamic, time-changing environments
- The **usual approaches** for querying, clustering and prediction use batch procedures and cannot cope with this streaming setting.
- Machine Learning **algorithms assume:**
  - Instances are independent and generated at random according to some probability distribution D.
  - It is required that D is stationary
- **Practice:** finite training sets, static models

Stream mining today

- We need to maintain decision models in real time.
- Learning algorithms must be capable of:
  - **Incorporate new information** at the speed data arrives
  - **Detect changes** and adapt decision models to the most recent information
  - **Forget outdated information**
- **Latest practice:** Unbounded training sets, dynamic models.

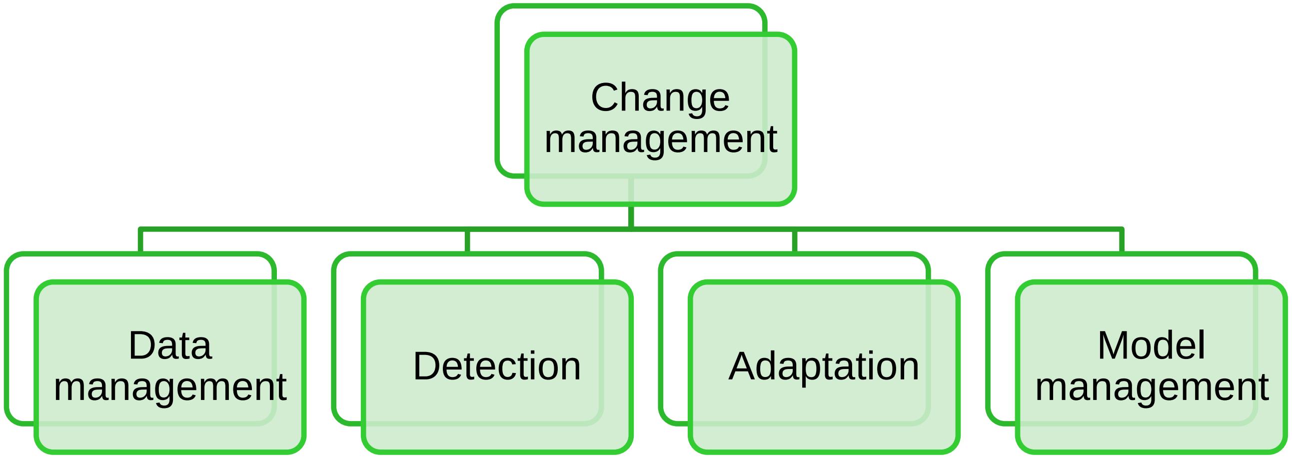
Gama J. Data Sciences: where are we going? Plenary speech @ IDEAL 2020

Terminology

- **DEF:** The **goal of change detection** is to monitor, detect and respond to changes in the systems which are modeled and in which decisions are made by computer-based models
  - **Change detection in streaming systems** is additionally complicated by the unbounded nature of the inputs and limited storage and processing power and the necessity to produce outputs in a timely (often real-time) manner
  - Change detection solutions must be able to differentiate **noise vs change**
  - **Persistence** → there is a consistent set of records following the changed distribution
- **DEF:** **Drift** is gradual change

RESPONDING TO CHANGE

Traditional change management



- Machine learning solutions try to find function f that maps input x to output y

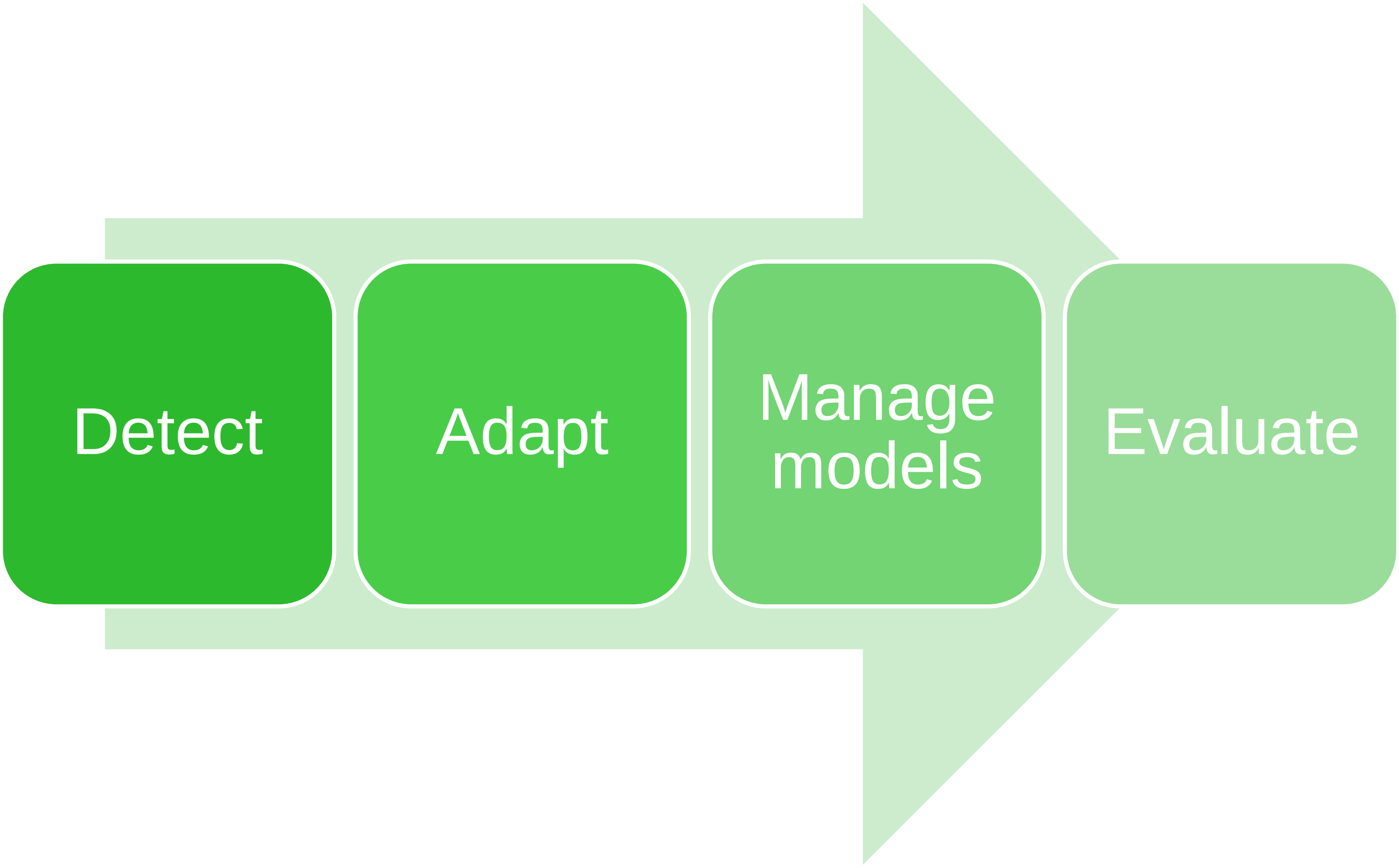
$$y = f(x)$$

- Such a function is assumed to be stationary, i.e. distribution generating the data is fixed (but unknown).
- Real-life datasets (and especially data streams) are non-stationary and evolving

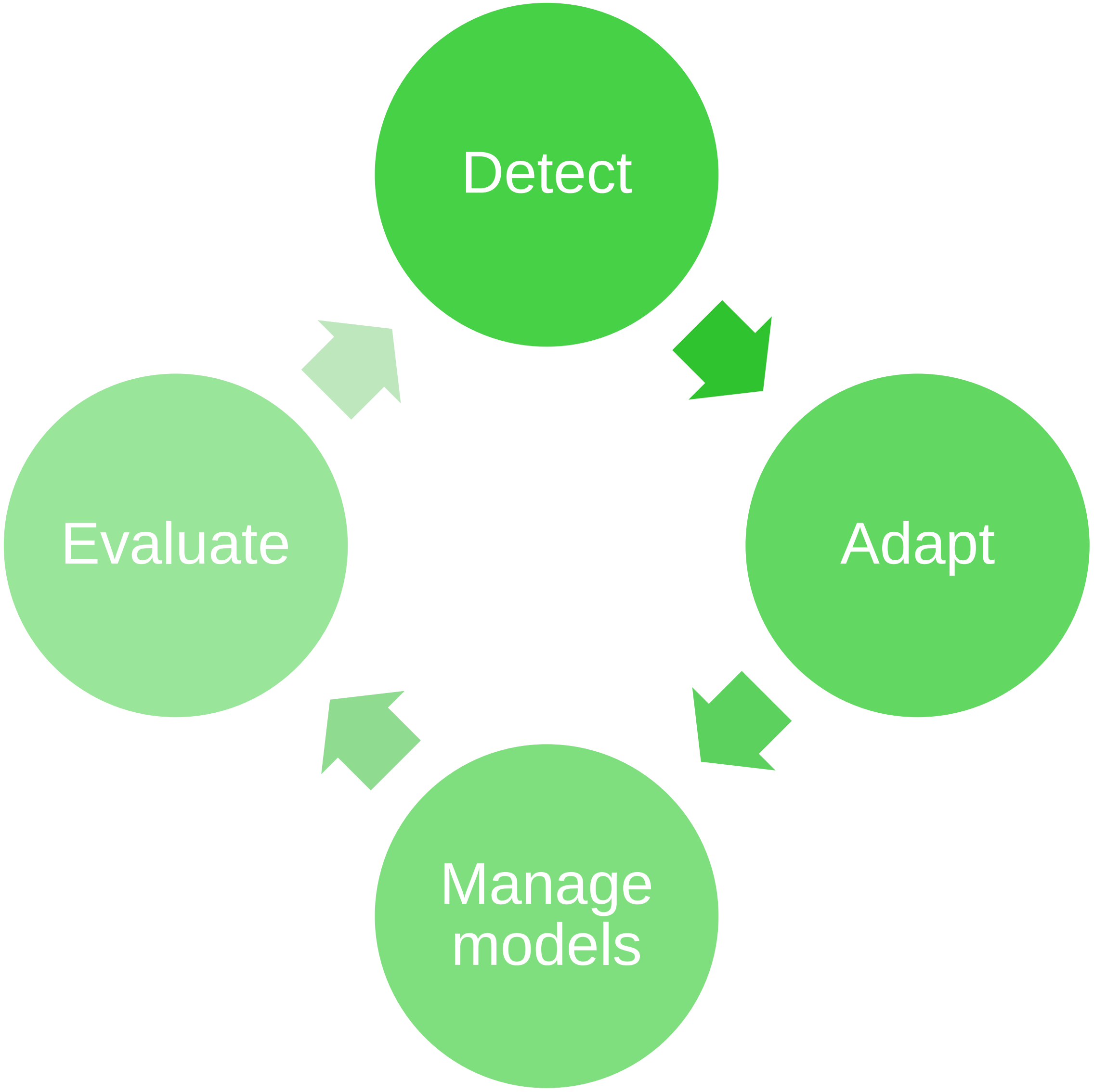
Change management concepts

- **DEF: Data management** techniques select and store the optimum type and amount of data and data aggregates for efficient change management
- **DEF: Change detection** algorithms monitor input data streams and machine learning solution performance with the goal to detect sub-optimal system operation
- **DEF: Adaptation** techniques enable the data analysis system to react (or preempt) changes
- **DEF: Model management** techniques allow the data analysis system to handle multiple models, train or re-train, retire
- **DEF:** Change detection **evaluation** techniques allow the data analysis system to measure its change management process → additional improvements, detect bugs

Change management process v1



Change management process v2

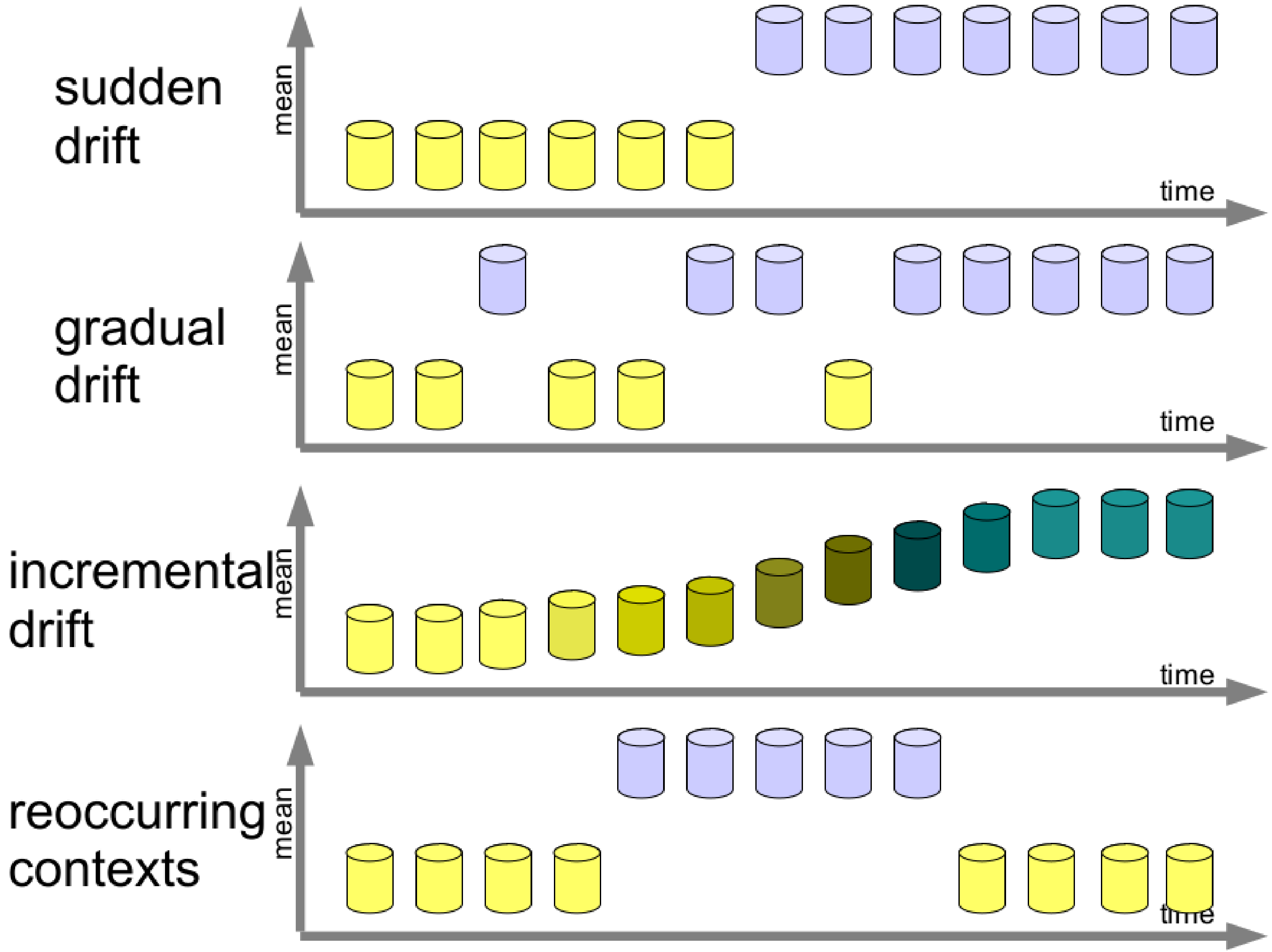


Phase I: Detect

Causes of change

- Changes due to modifications in the **context of learning**
  - The world changes → our assumption and system models need to be updated
  - New, improved techniques appear → change model
- Changes in **hidden variables**
  - ML algorithms learn from observations described by a finite set of attributes.
  - In real world problems, there can be important properties of the domain that are not observed → hidden variables
  - Hidden variables may change over time (!) → such changes can invalidate otherwise well-trained models
- Changes in the **characteristic properties** in the observed variables
  - The observed variables change in time → re-train necessary

Rate of change



Rate of change in words

- **Concept shift** usually refers to abrupt changes → usually 'easier' to detect
  - Re-occurring contexts are essentially periodically or otherwise repeated sudden drifts
- **Concept or gradual drift** is usually associated with gradual change in the observed concept(s)
  - Usually more challenging to detect compared to sudden drift
  - The initial phase of gradual change can be mistaken for noise in the incoming data
  - Resilience to noise can be obtained via observing more records
  - Incremental drift is gradual change with a 'smooth' transition

- **Note:** if the rate of change is larger than the ability to learn, then it is not possible to set up a proper machine learning solution

Types of change

Change Detection 37

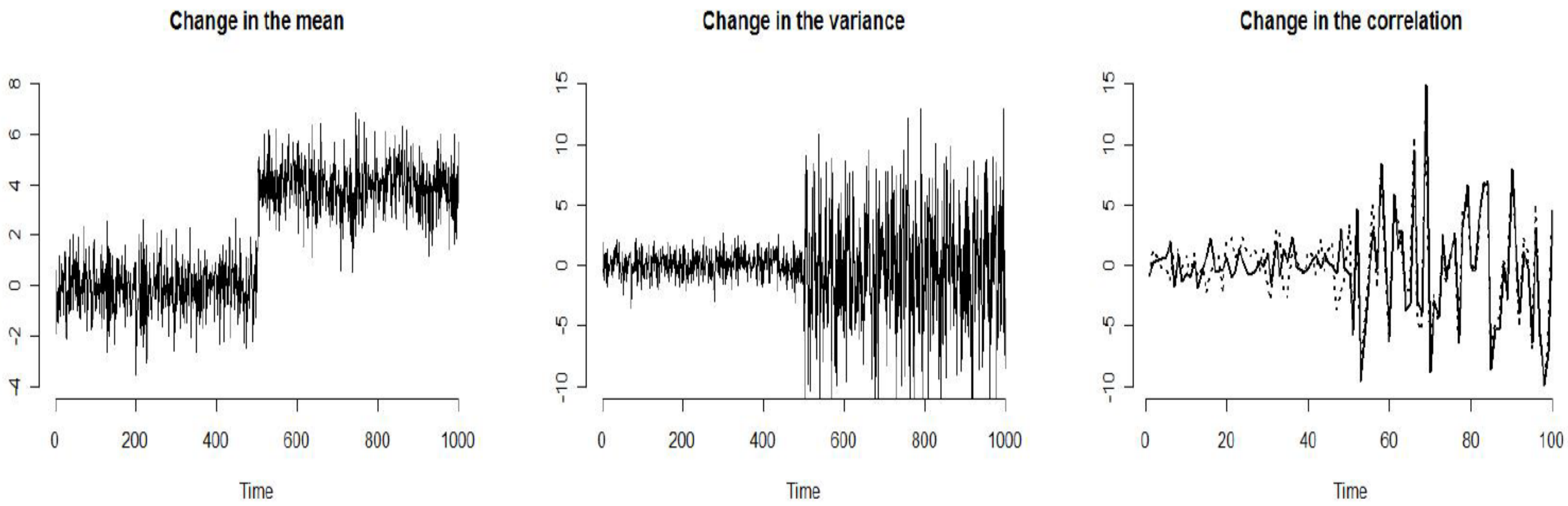


Figure 3.1: Three illustrative examples of change: (a) Change on the mean; (b) Change on variance; (c) Change on correlation

Gama J. (2010). Knowledge discovery in data streams. CRC Press.

Data management approaches

- Change detection techniques rely on data
- The data management approaches used by such techniques can be categorized into
  - **Full memory** → store statistics over all examples → their implementation in streaming systems can be challenging
    - Usually rely on some form of sketches (stream aggregates)
    - Example: weighted average over all observations
  - **Partial memory** → store only the most recent observations
    - **Fixed window size** → store in memory a fixed amount of the most recent observations
    - **Adaptive window size** → usually works with ‘longer’ windows in normal operation and with ‘shorter’ windows when change is detected
    - Example: 20 SMA during normal operation and 10 SMA when change is suspected to come
- Weigthing observations in the data management solution allow gradual forgetting of ‘older’ observations

Monitoring in detection

- Monitoring the evolution of **performance indicators** of the decision model
  - Usually monitoring accuracy, recall and precision over time and raise the red flag (i.e. signal change) when the values of these indicators are outside some predefined (or varying?) bound(s)
  - Most change detection approaches fall into this category
- Monitoring data distributions on **two different time-windows**
  - Usually summarize and compare past information and the most recent observations
  - One approach is to examine observations drawn from two (or more?) probability distributions and decide whether they are different

Cumulative sum (CUSUM)

- CUSUM (CUmulative SUM) algorithm is a change detection algorithm which monitors the cumulative sum to detect a change.
- **DEF:** Let S_t be the current cumsum and m_t the current min value of S_t, the cumsum compares this difference with a threshold.

$$z_t = (x_t - \mu) / \sigma$$
$$S_0 = 0$$
$$S_t = \max(0, S_{t-1} + z_t - k)$$

Declare change if

$$S_t > h$$

Reset CUSUM by:

$$S_0 = 0, reset\ \mu, \sigma$$

- **Challenge:** how to choose values k & h?
  - Guideline for k: set it to half the value of the change to be detected (measured in standard deviations)
  - Guideline for h: set $h = \ln(\frac{1}{\delta})$ where δ is the acceptable false alarm rate

Page-Hinkley (PH) test

- The Page-Hinkley test is essentially a variant of CUSUM

$$z_t = (x_t - \mu) \,/\, \sigma$$

$$s_0 = 0$$

$$s_t = s_{t-1} + z_t - k$$

$$S_t = min\{s_t, S_{t-1}\}$$

Declare change if

$$g_t - G_t > h$$

Reset PH to

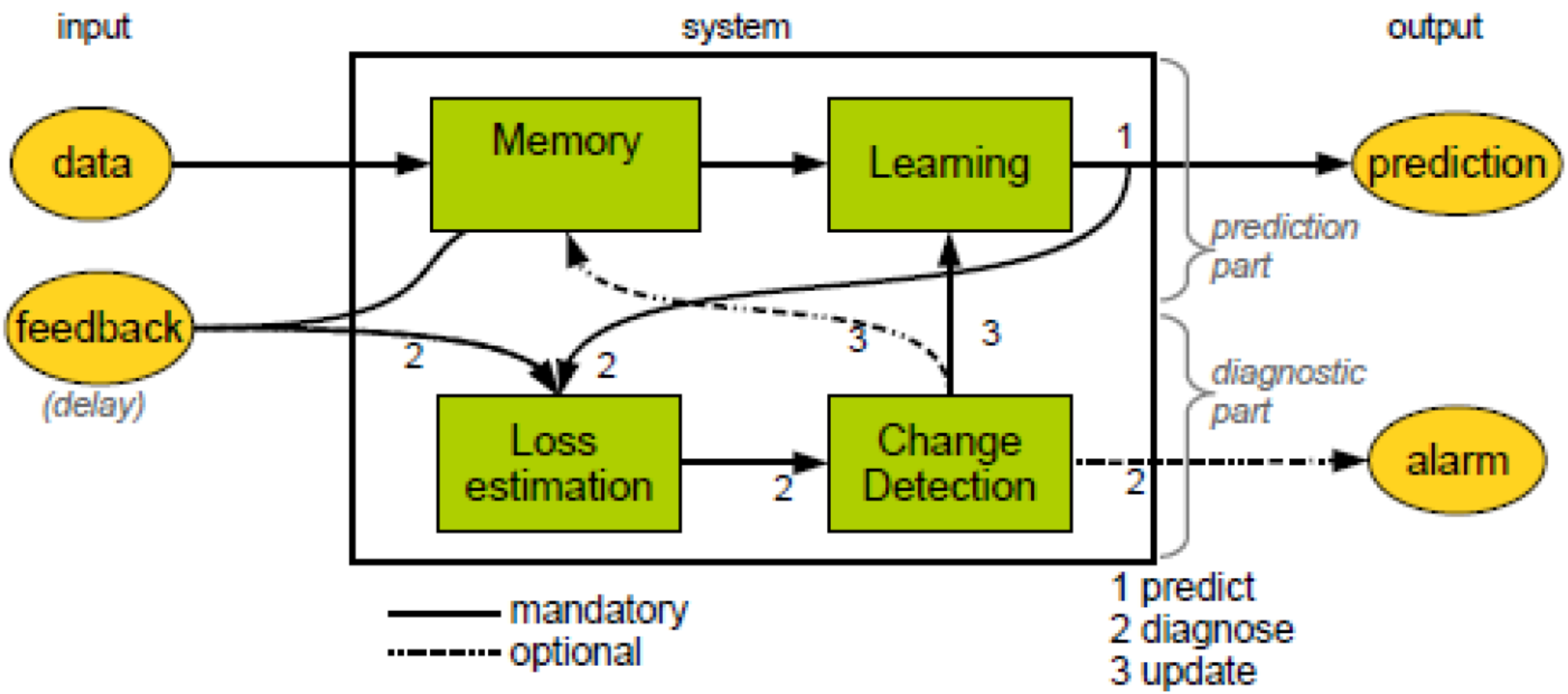
$$S_0 = 0, reset\ S_t, \mu, \sigma$$

Phase II: Adapt

Adaptation methods

- **Blind methods** adapt (i.e. modify) decision models at regular intervals without considering whether changes have really occurred.
  - Examples include methods that weight the examples according to their age and methods that use time-windows of fixed size
  - **Pro:** no detection is necessary
  - **Contra:** re-train might not be necessary → wasted resources
- **Informed methods** only modify the decision model after a change was detected.
  - They are used in conjunction with a detection model
  - **Pro:** optimized for resource use → not run when not needed
  - **Contra:** might miss changes due to noise or delays

Adaptive learning solution



A generic schema for an online adaptive learning algorithm.

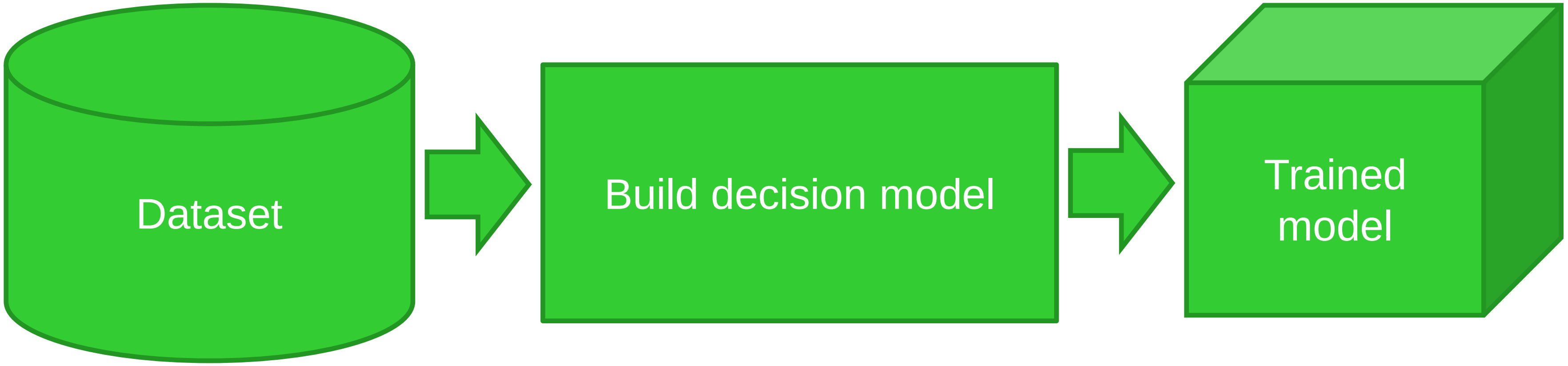
(*A survey on concept drift adaptation*, J.Gama et al, ACM-CSUR 2014)

PHASE III: MANAGE MODELS

Manage models

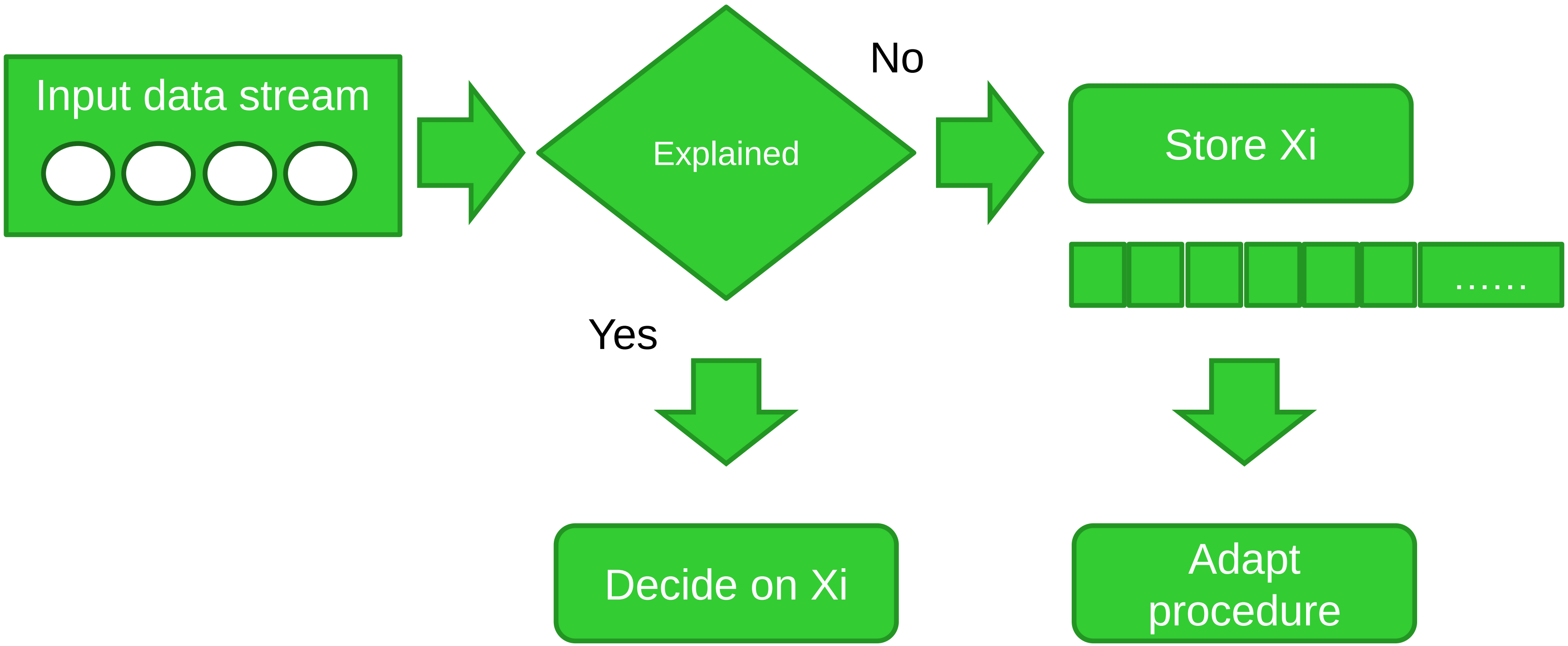
- Instead of maintaining a single decision model often it is optimal to use multiple decision models
  - This is different from ensemble models used to obtain optimal decisions in different ML use cases (!)
- **DEF:** The task of the **model management phase** is to store (in memory or in storage), archive and retire models
  - The model building is triggered in the detect phase
  - The models are built in the adapt phase
- Model management phases:
  - Phase I (offline): A model is built offline
  - Phase II (online): The detect and adapt phases yield in new models as well as models which should be retired

Offline training phase



- Build a decision model for known normal patterns
- The MM phase is tasked to store the trained model – at the right in the above figure

MM in the online adaptation phase



- **Step #1:** Detect observations or sets of observations for which a consistent decision cannot be made → store them in short-term memory
- **Step #2:** Short-term memory full → initiate adaptation procedure
- **Step #3:** Manage models, e.g. add newly trained and retire old model

Phase IV: Evaluate

Evaluate change detection

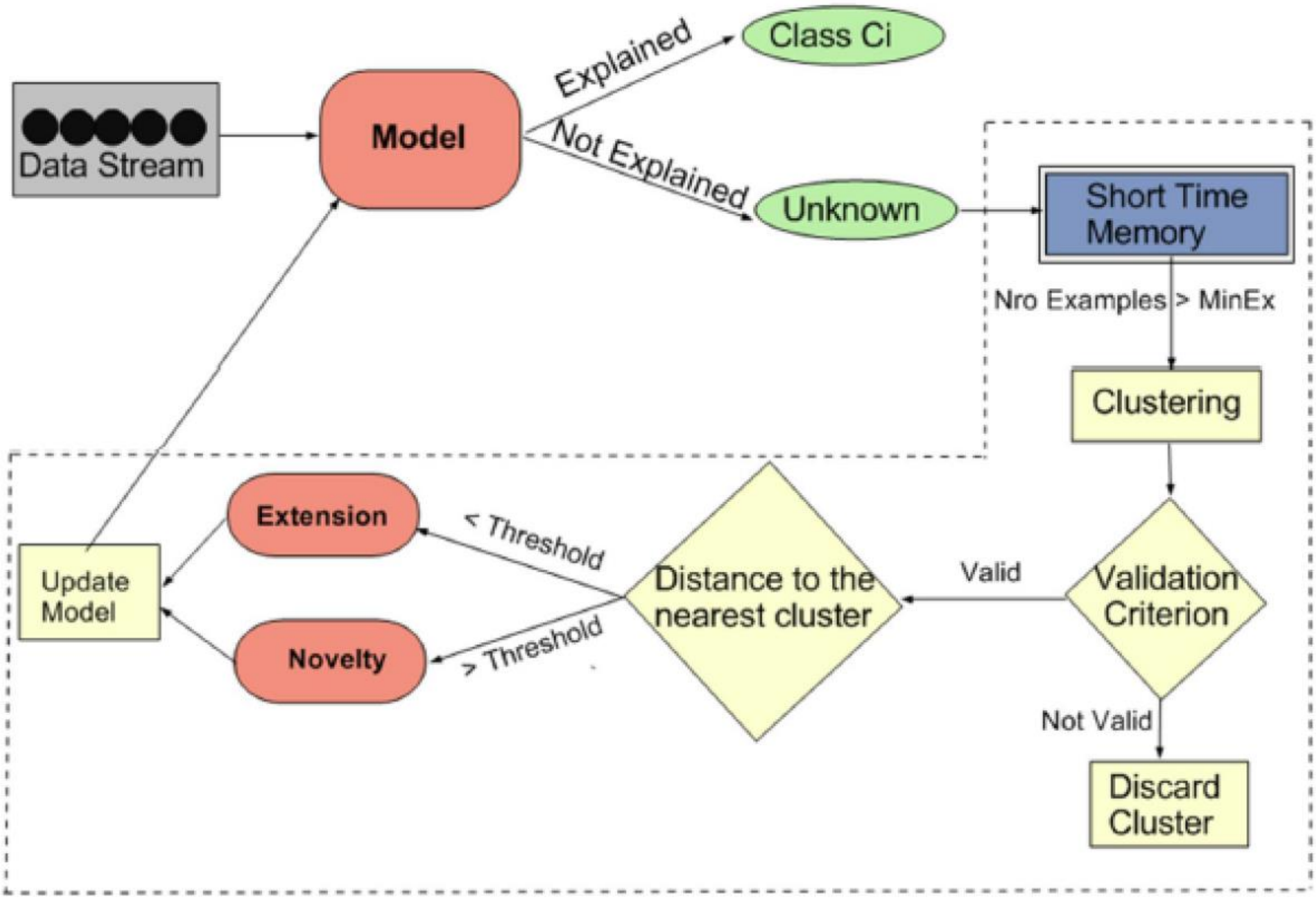
- Drift detection methods are evaluated on the following metrics:
  - **Error rate** (Number of mistakes made so far) → possible if there is (delayed) feedback about changes, i.e. they are labeled
  - **Probability of true detection** or TPR → capacity to detect and react to change
  - **Probability of false alarm** or FPR → does not signal when there is no change in the observed concept
  - **Delay in detection** → usually measured as the number of examples required to detect a change after the occurrence of a change
  - Other possible performance indicators: precision, recall, AUC etc.

CHANGE DETECTION USE CASES

Change detection use cases

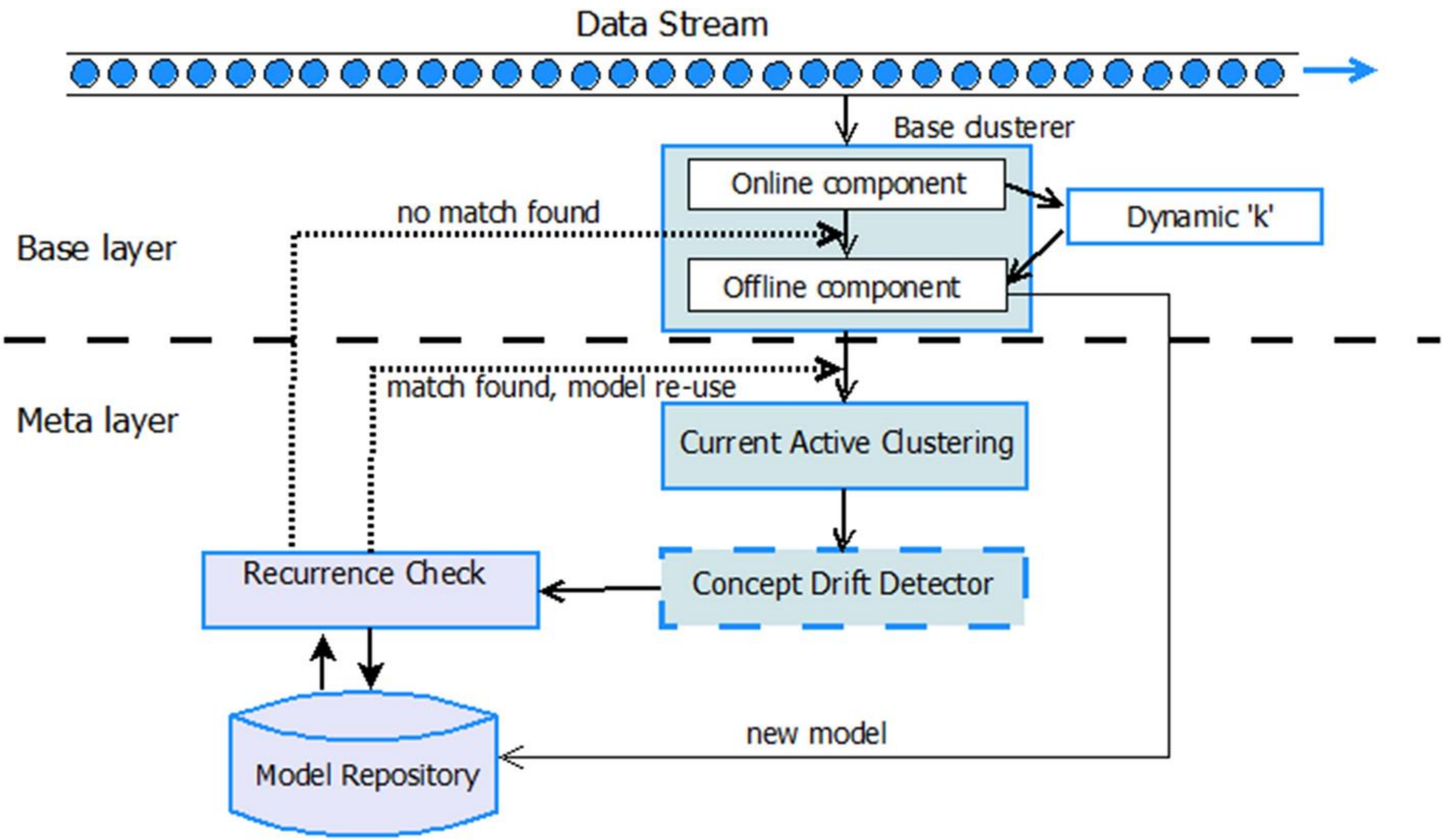
- **Social media** → sentiment analysis, e.g. via detecting changes in Twitter feeds → the analysis of data streams consisting of short texts
- **Information security**
  - Changing behavior of employees → more complex insider threat monitoring
  - Changing behavior of adversaries → frequent adaptation to the latest threats, e.g. based on security data feeds, intra-industry collaboration
- **Financial systems**
  - Macro-level market changes → re-visit models used for forecasting market movement(s)
  - Typical residential and commercial customer behavior shifts → use different solutions to predict churn, upsell, analyze (credit) risk
- **Health infrastructure**
  - Changing habits of the population → changing health of average male and female patients
- **Electric power systems**
  - Residential, commercial and industrial blocks are re-purposed, customers install energy saving tech → changing power consumption patterns

Classification in the presence of change



Gama J. Data Sciences: where are we going? IDEAL 2020.

Clustering in the presence of change



Namitha K., Santhosh Kumar G. (2020). Learning in the presence of concept recurrence in data stream clustering, vol 7 (75).

Prediction in the presence of change

- The Drift Detection Method (DDM) is applicable in the context of predictive models
  - The method monitors the number of errors produced by a model learned on the previous stream items
  - When DDM observes that the prediction error increases, it takes this as evidence that change has occurred

$$s_t = \sqrt{p_t(1 - p_t)/t},$$
$$p_t \; is \; the \; error \; rate$$

- DDM stores the smallest error rate p_{min} observed up until moment t and performs the following checks
  - Issue warning if $p_t + s_t \geq p_{min} + 2 \cdot s_{min}$
  - Declare change if $p_t + s_t \geq p_{min} + 3 \cdot s_{min}$
- **DDM drawback:** p_t is computed based on all observations since the last change → DDM might be slow to respond

Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004, September). Learning with drift detection. In Brazilian symposium on artificial intelligence (pp. 286-295).

Anomaly detection in the presence of change

- Open discussion with students

- Example questions for discussion:
  - Which datasets are we analyzing?
  - What do we want to find out, i.e. what are the anomalies (outliers) we are interested in?
  - How to detect change?
  - Classification vs clustering? Or other techniques?
  - Are motifs and discords useful in the changing anomaly detection scenario in unbounded data streams?
  - Is windowing necessary? Event time vs processing time windowing?

REFERENCES

References

- Barddal J.P., Gomes H.M., Enembreck F. (2015). Advances on Concept Drift Detection in Regression Tasks Using Social Networks Theory, International Journal of Natural Computing Research, vol 5(1).
- Bifet, A., Gavaldà, R., Holmes, G., & Pfahringer, B. (2018). Machine learning for data streams: with practical examples in MOA. MIT Press.
- Bifet, A., Holmes, G., Pfahringer, B., & Gavaldà, R. (2011, August). Mining frequent closed graphs on evolving data streams. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 591-599).
- Demšar, J., & Bosnić, Z. (2018). Detecting concept drift in data streams using model explanation. Expert Systems with Applications, 92, 546-559.
- Gama J. Data Sciences: where are we going? (plenary talk) (2020). 21st International Conference on Intelligent Data Engineering and Automated Learning - IDEAL'2020 4-6 November Guimaraes, Portugal
- Gama J. (2010). Knowledge discovery in data streams. CRC Press.
- Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004, September). Learning with drift detection. In Brazilian symposium on artificial intelligence (pp. 286-295). Springer, Berlin, Heidelberg.
- Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM computing surveys (CSUR), 46(4), 1-37.
- Janardan, S.M. (2017). Concept drift in Streaming Data Classification: Algorithms, Platforms and Issues. Procedia Computer Science, vol 122, pp. 804-811.
- Namitha K., Santhosh Kumar G. (2020). Learning in the presence of concept recurrence in data stream clustering, vol 7 (75).
- Page E.S. (1954). Continuous inspection schemes. Biometrika, 41(1/2):100–115.

Thank you for your attention!