Research and Innovation Center of Excellence

MSc on Intelligent Critical Infrastructure Systems

Machine Learning

Lecture 8: Online Learning

Kleanthis Malialis
Research Associate
KIOS Research and Innovation Center of Excellence
University of Cyprus

University of Cyprus

Imperial College London

FUNDED BY:

Course outline

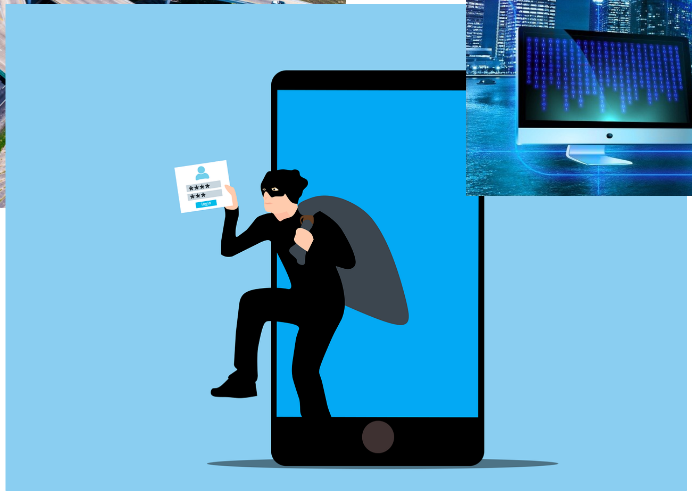
- **Week 1**
  - Introduction and Preliminaries
- **Week 2**
  - Linear Regression
  - Regularisation, Logistic Regression, SVMs
- **Week 3**
  - Neural Networks and Deep Learning
- **Week 4**
  - Feature Engineering and Evaluation
  - Online Learning
- **Week 5**
  - Unsupervised Learning
- **Week 6**
  - Reinforcement Learning
- **Week 7**
  - Monitoring and Control

Motivation

- An ever-increasing volume of data is nowadays becoming available in an **online** fashion, in various real-world applications:

Water networks

Transportation networks



Finance

Security

Social media

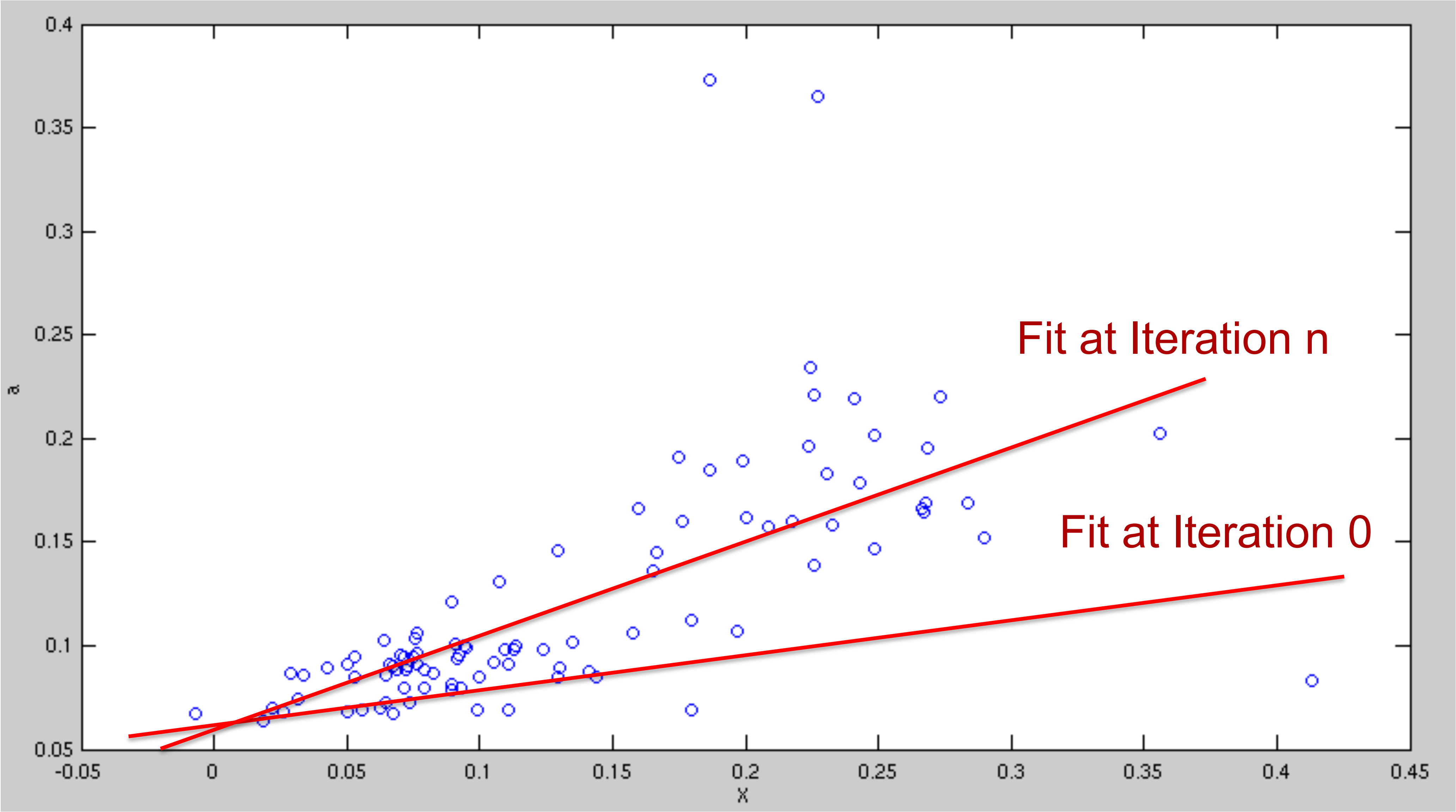
There is at present an emerging need where predictive models are **trained on-the-fly** as new information becomes available.

Online learning provides **adaptation capabilities**, necessary to **maintain optimality**.

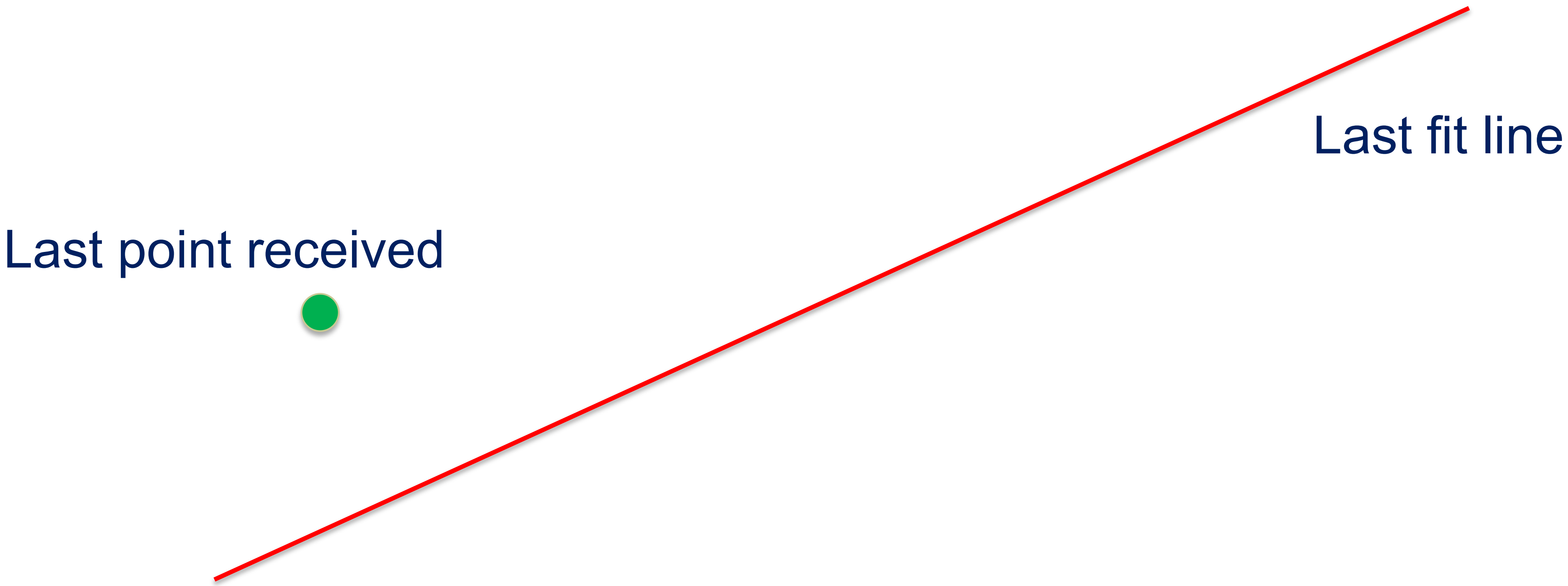
Online learning

- In **offline learning**, data is collected for some time (batch) and then machine learning algorithms are applied on the batch data.
- In **online learning**, training occurs in consecutive rounds. At the beginning of each round, the algorithm is presented with an input sample, based on which it makes a prediction. Based on the difference between the prediction and the desired/true output, the model is adapted for subsequent rounds.
- Online learning doesn't necessarily mean streaming data, but usually it is applied to streaming data. Sometimes, even called **stream learning**.

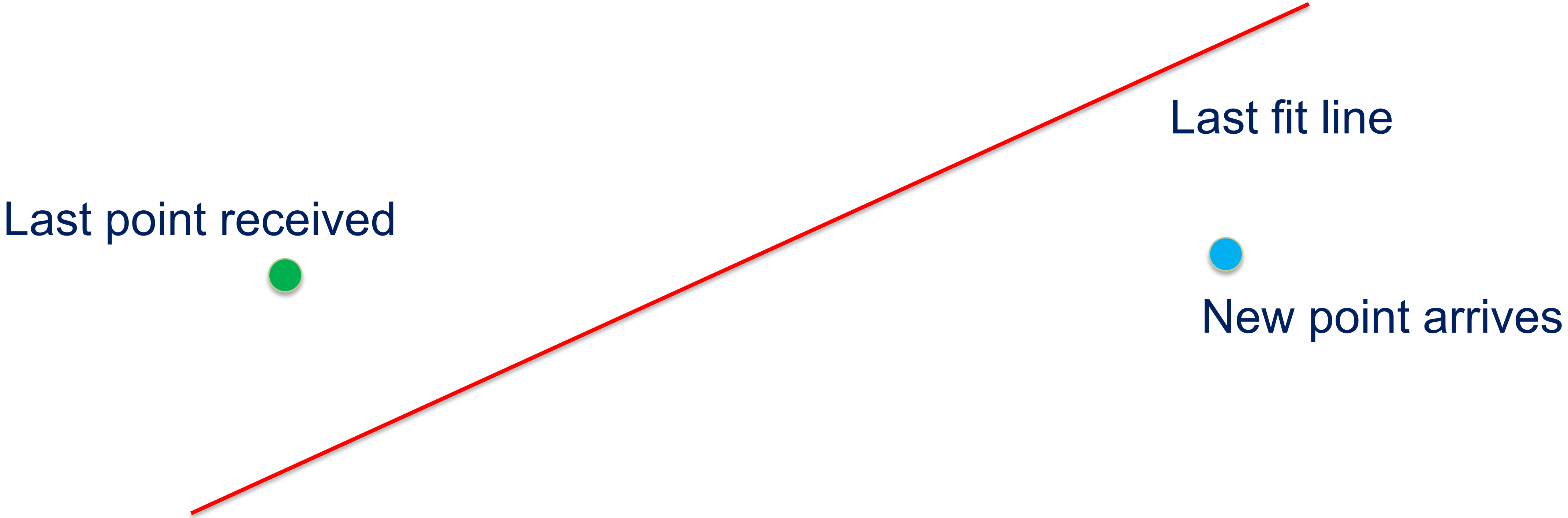
Linear Regression



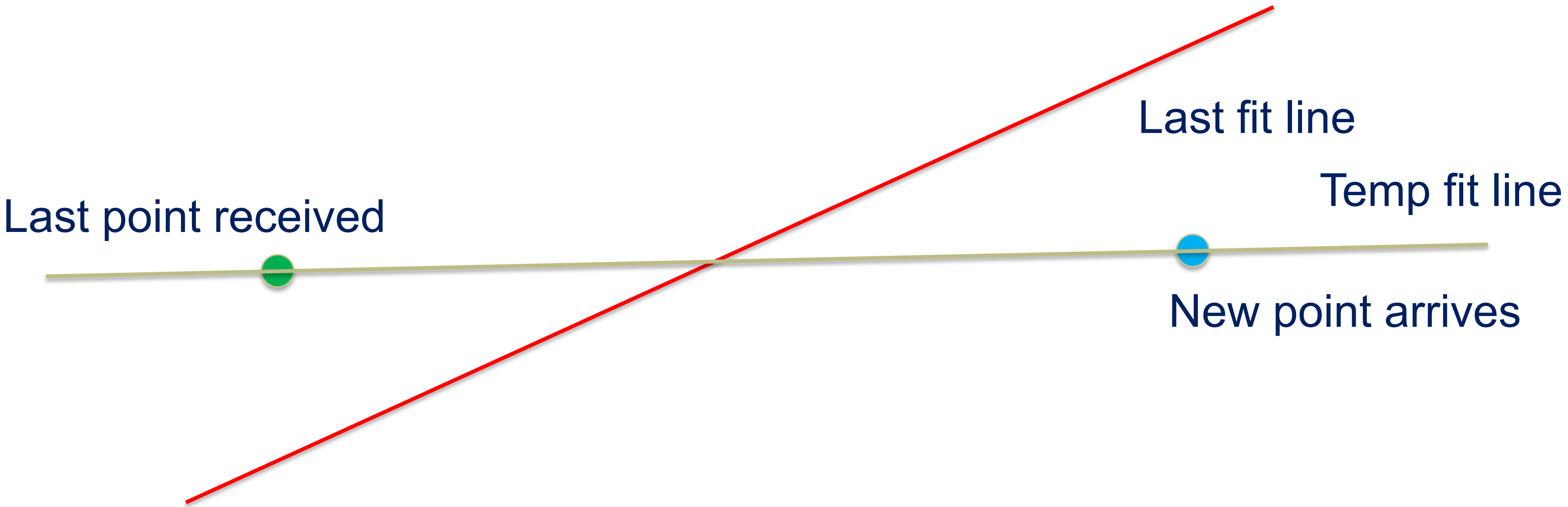
Online Linear Regression



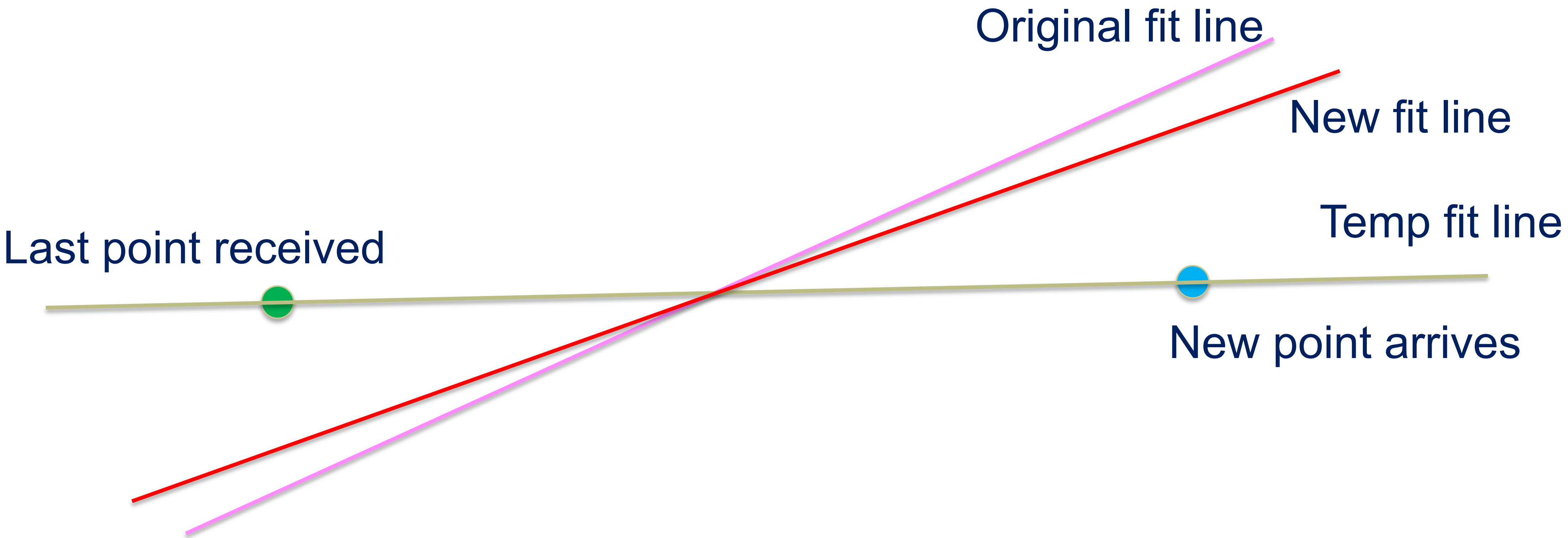
Online Linear Regression



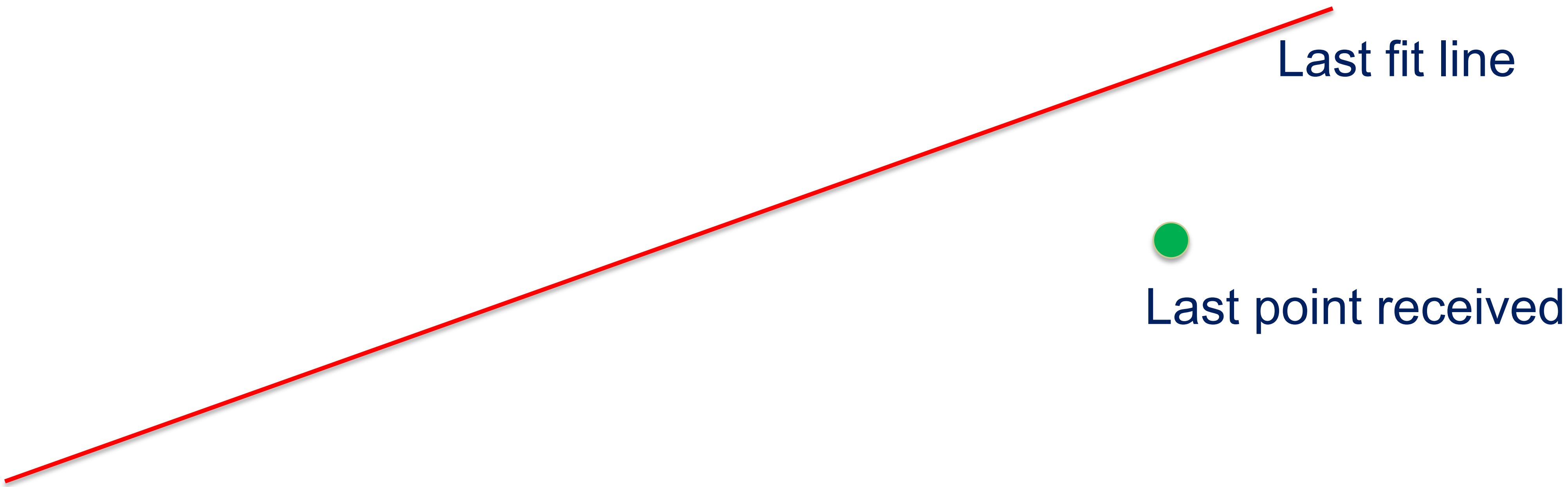
Online Linear Regression



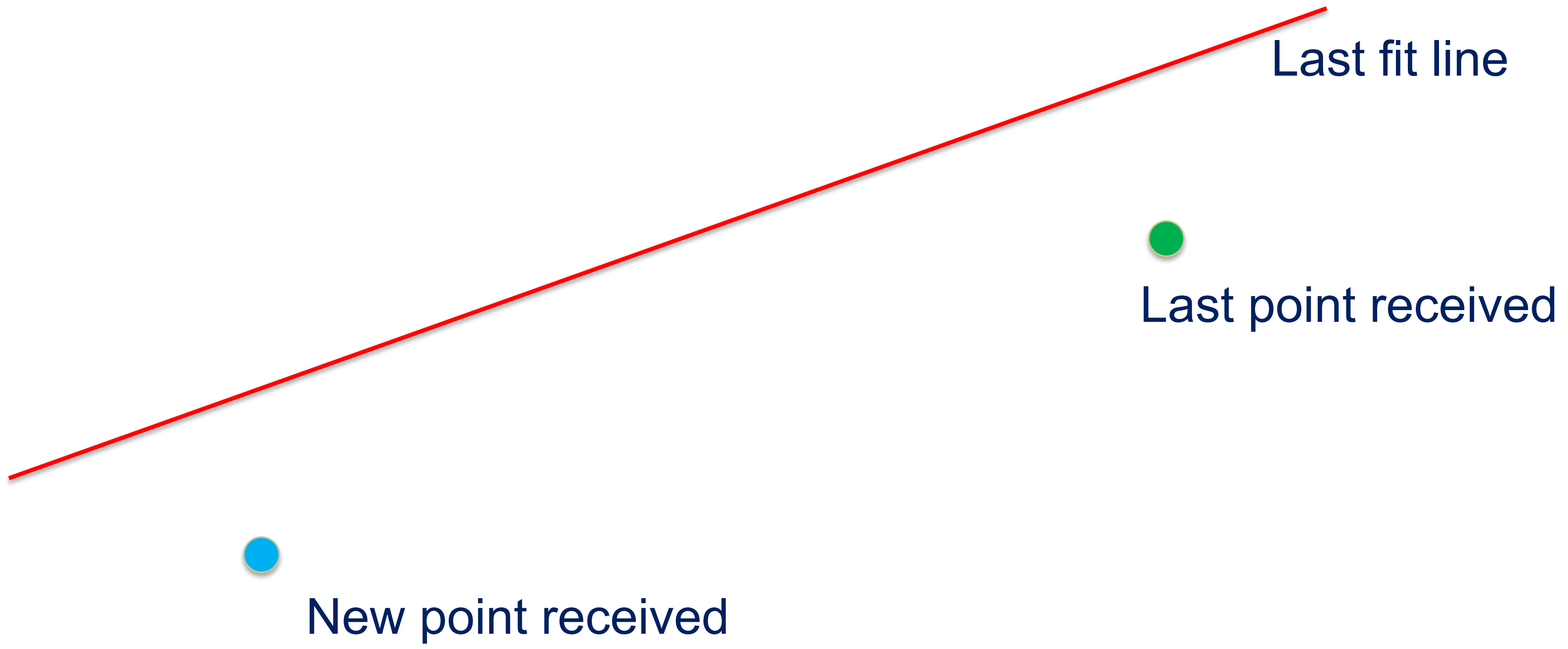
Online Linear Regression



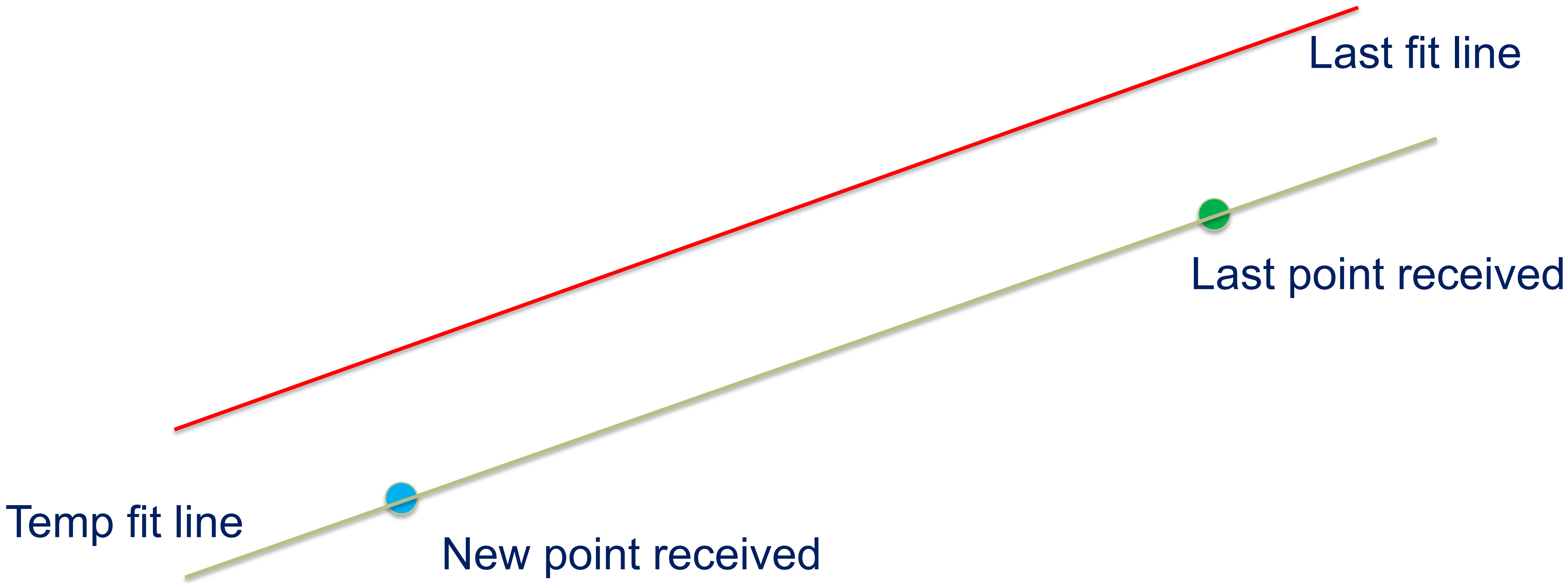
Online Linear Regression



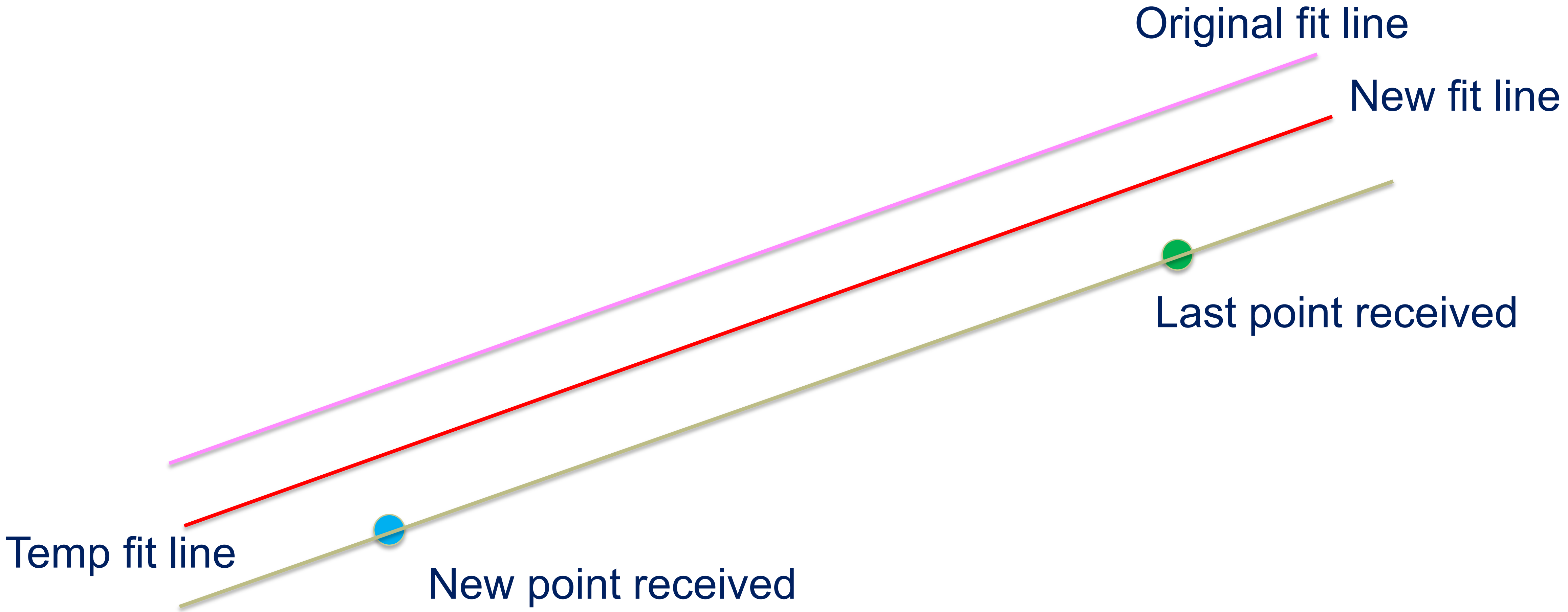
Online Linear Regression



Online Linear Regression



Online Linear Regression



Online learning: Desired properties

1. **Learning new knowledge**
   - As new information is becoming available.
2. **Preserving previous knowledge**
   - What previous knowledge is still relevant (hence, to preserve it), and what has now become irrelevant (hence, to “forget” it).
3. **High performance**
   - On both majority and minority classes.
4. **Fast operation**
   - It should operate before the arrival of the next example.
5. **Fixed memory**
   - Ideally, no memory!

Challenging to obtain a trade-off!

Online learning framework

For time $k = 1, 2, \ldots$

1. receive question (input) $\mathrm{x}^{\mathrm{k}} \in X$
2. predict answer (output) $\hat{y}^k \in \hat{Y}$
3. receive true answer $\mathrm{y}^{\mathrm{k}} \in Y$
4. calculate loss $L(y^k, \hat{y}^k)$
5. update the predictive model
6. discard x^{k}

online supervised learning

incremental learning
$f^k = f^{k-1}.train()$

one-pass learning
(but see Property 2)

Applications

- Fraud detection
- Spam detection
- Financial portfolio selection
- Online ad placement
- Online web banking
- Real-time monitoring
- Navigation and control

Online Stochastic Gradient Descent (SGD) for Linear Regression

For time $k = 1, 2, \ldots$

1. receive question (input) x^{k}
2. predict answer (output) $\hat{y}^{k} = sigm((\theta^{t-1})^{T} x^{t})$
3. receive true answer y^{k}
4. calculate logistic loss $L\left(y^{k}, \hat{y}^{k}\right) = -y^{t} \log(\hat{y}^{t}) - (1 - y^{t}) \log(1 - \hat{y}^{t})$
5. update the classifier $\theta_{j}^{\mathrm{t}} = \theta_{j}^{t-1} - \alpha \frac{\partial}{\partial \theta_{j}^{t-1}} L(\theta)$
6. discard x^{k}

Online learning: Challenges

- Nonstationary targets
- Class imbalance
- Limited supervision

Non-stationary targets

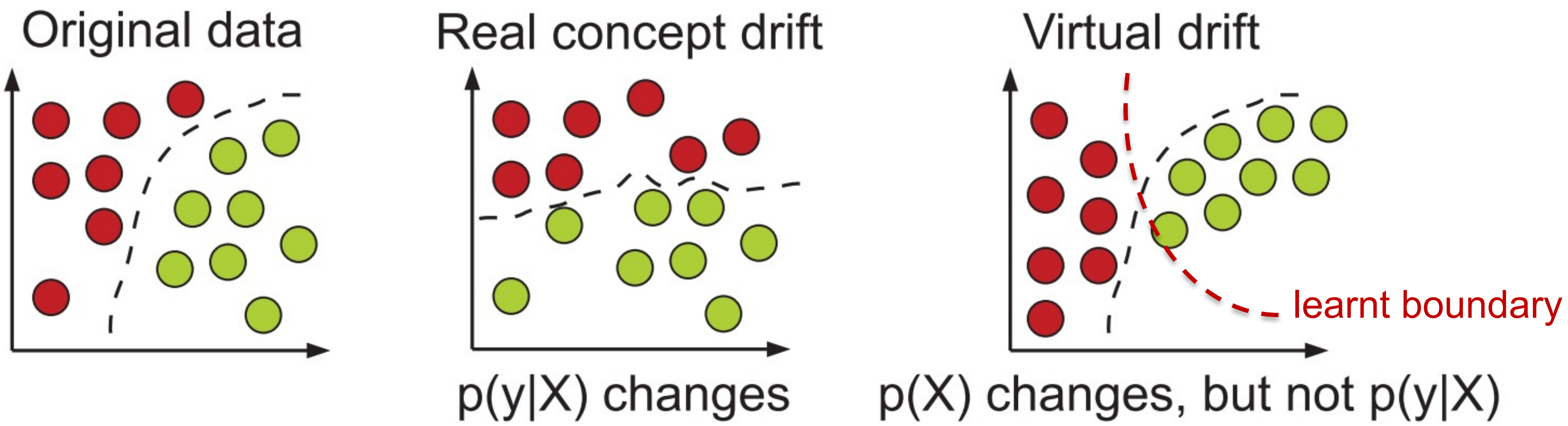
The target function that we are trying to learn may be stationary or dynamic.

- **Stationary targets**. The target function we are trying to learn does not change over time, but it is unknown (or uncertain) and it may be stochastic.
- **Non-stationary (time-varying) targets**. The target function we are trying to learn not only it is unknown, but it is changing over time. It may even be adapting to our model (for example, in an adversarial manner).

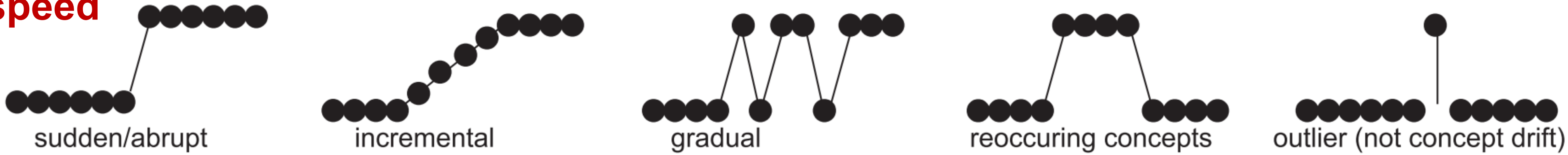
Nonstationary targets: Concept drift

Concept drift refers to a change in the joint probability: $\exists X : p_{t_0}(X, y) \neq p_{t_1}(X, y)$

type



speed



Concept drift adaptation

Passive methods

- They implicitly address the problem by continually updating a learning model using **incremental learning**.
- **Memory-based**, e.g., Sliding window, adaptive window, multiple windows.
- **Ensembling**

Hybrid

Active methods

- They explicitly address the problem by using drift detection. Typically, they discard a model and perform a **complete re-training** when a drift alarm is triggered.
- **Statistical tests**, e.g., binomial distribution hypothesis testing.
- **Threshold-based**

Concept drift adaptation: Memory-based

- They typically employ a **sliding window** to maintain a set of recent examples that a learning algorithm is (incrementally) trained on.

- **Challenge**: Determine a priori the window size.
  - A larger window is better suited for gradual drift, while a smaller window is suitable for an abrupt drift.

- **Solutions**:
  - Adaptive sliding window
  - Multiple sliding windows

- **Note**:
  - No longer one-pass learning

Concept drift adaptation: Ensembling

- **Idea**
  - An ensemble of classifiers can improve performance and provide the flexibility of injecting new data by adding classifiers or “forgetting” irrelevant data by removing or updating existing classifiers.
- **Algorithms**
  - Weighted Majority algorithm

Tutorial!

Concept drift adaptation: Threshold-based

1. Start with a pre-trained model (or wait for some time)
2. Set a reference window, calculate the average loss $\boldsymbol{avg_r}$, and set a threshold $\boldsymbol{\theta_{alarm}}$.
3. Have a moving window, and continually monitoring the average loss $\boldsymbol{avg_k}$ for a decrease in performance
4. Re-train when necessary: $\boldsymbol{avg_r - avg_k > \theta_{alarm}}$
5. Repeat

Reference window — Moving window

$$\boxed{c^1 \quad c^2 \quad \dots \quad c^{10}} \quad \dots \quad \boxed{c^{k-9} \quad c^{k-8} \quad \dots \quad c^k}$$

$$x^1 \quad x^2 \quad \dots \quad x^{10} \quad \dots \quad x^{k-9} \quad x^{k-8} \quad \dots \quad x^k$$

$$c^k = \begin{cases} 0, if\ y^k \neq \hat{y}^k \\ 1, if\ y^k = \hat{y}^k \end{cases}$$

Learning paradigms for limited supervision

- Online active learning
- Online unsupervised learning

Active learning

- It is concerned with strategies to selectively query for class labels from an **oracle** (typically, a human expert), based on a **budget** $B \in [0,1]$.

- Several industrial large-scale classification systems have been realised through AL:

Labelling malicious ads

Google Ads

Autonomous vehicles with self-driving capabilities

TESLA

Online active learning

For time $k = 1, 2, \ldots$

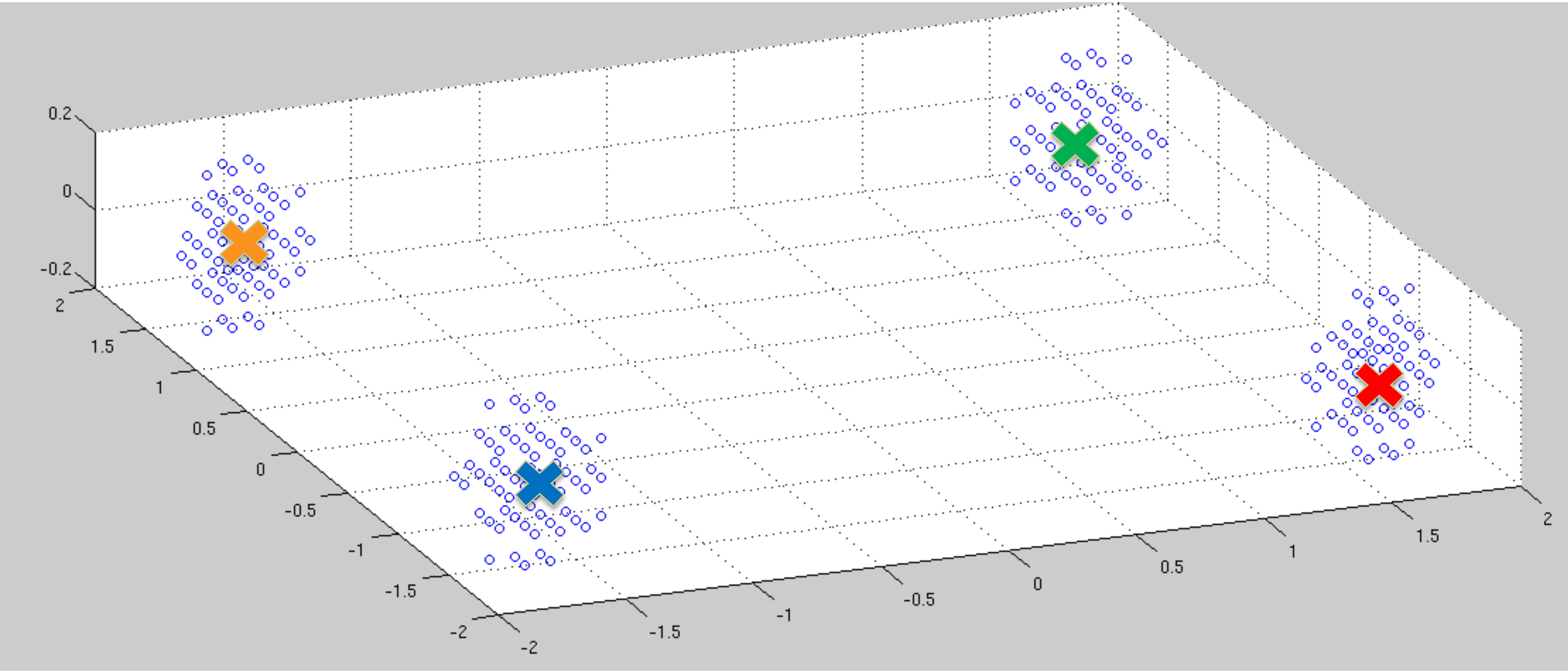
1. receive question (input) $\mathrm{x}^{\mathrm{k}} \in X$
2. predict answer (output) $\hat{y}^k \in \hat{Y}$
3. If budget allows & *AL_strategy* == *True*:
   1. receive true answer $\mathrm{y}^{\mathrm{k}} \in Y$
   2. calculate loss $L(y^k, \hat{y}^k)$
   3. update the predictive model
   4. discard x^{k}
4. update budget

active learning strategy

Uncertainty sampling AL strategy

- Requests the label of the most uncertain instance.
- Most common active learning strategy.
- Let $f(x^k) = max_y\, \hat{p}(y|x^k)$ be the best prediction probability.
- **Fixed uncertainty sampling**: $f(x^k) < \theta$
- **Randomised variable uncertainty sampling**
  - Variability: $$\theta = \begin{cases} \theta(1 - s) & if\ f(x^k) < \theta_{rdm} \\ \theta(1 + s) & if\ f(x^k) \geq \theta_{rdm} \end{cases}$$
  - Randomisation: $\theta_{rdm} = \theta \times \eta,\ \eta \sim N(1, \delta)$

Clustering



Online Clustering

Step k-1

Online Clustering

Step k

Online Clustering

Step k

Online Clustering

Step k