KIOS
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**
  - Advanced topics & Applications

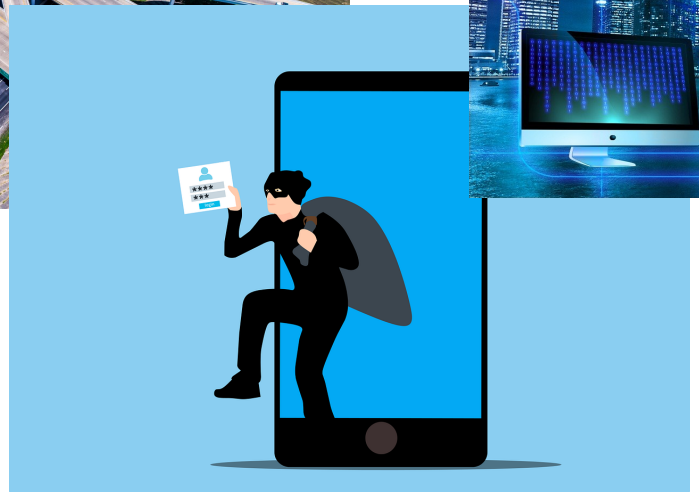
- **Assignments**
  - 31/03 – 11/04
  - 08/04 – 15/04
  - 15/04 – 22/04

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

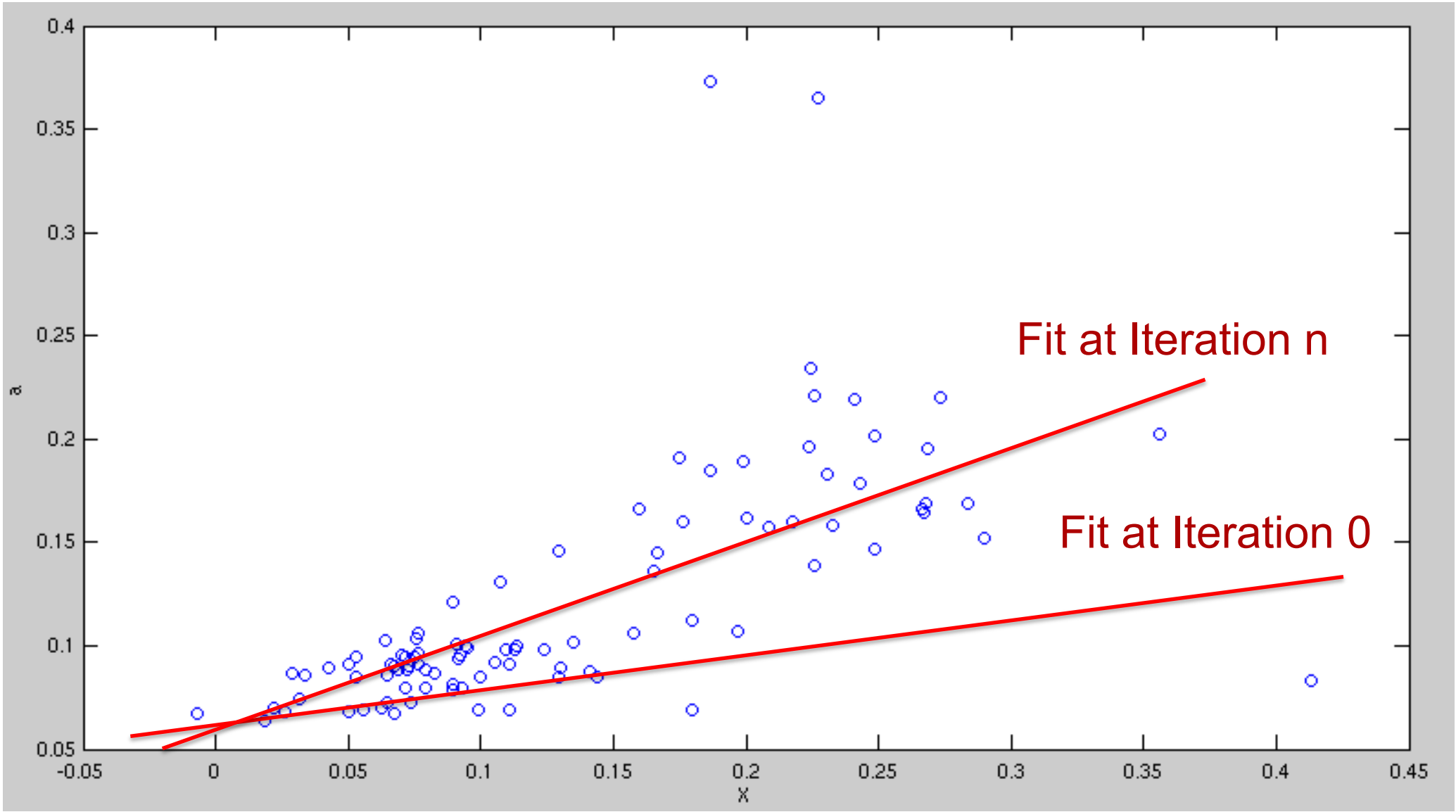
There is at present an emerging need where predictive models are **trained on-the-fly** as new information becomes available.

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

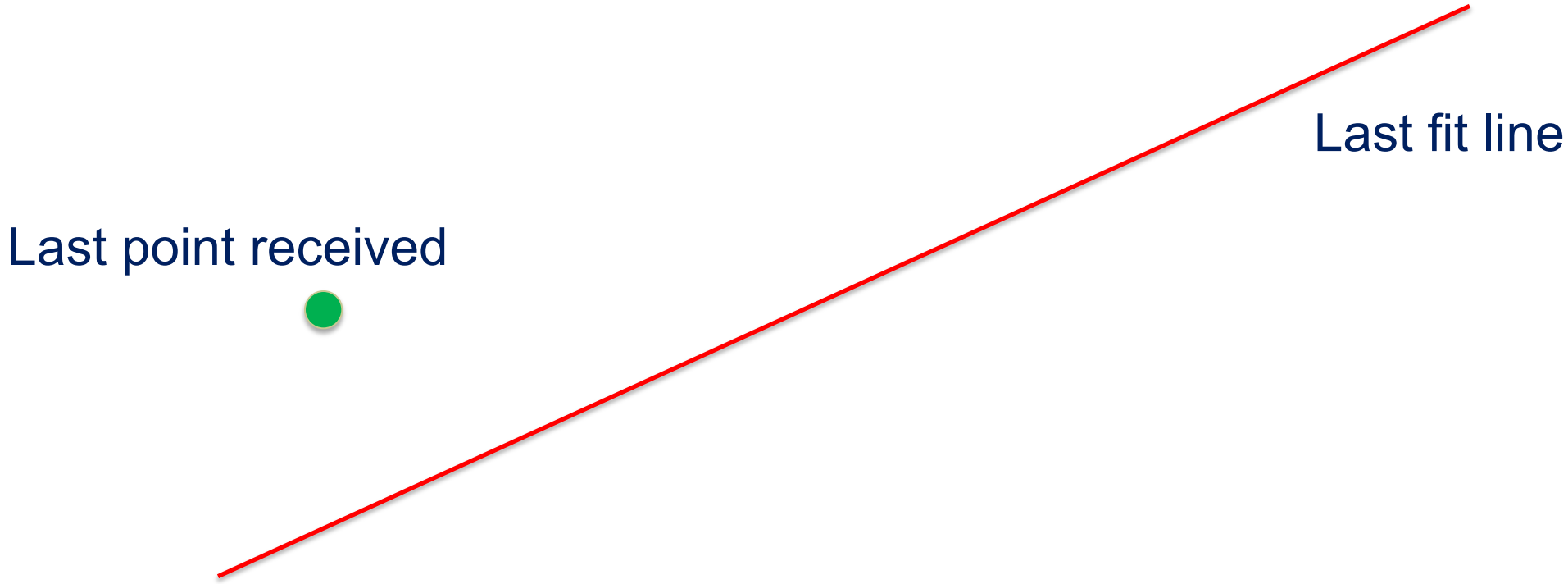
Online learning

- 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 **streaming learning**.

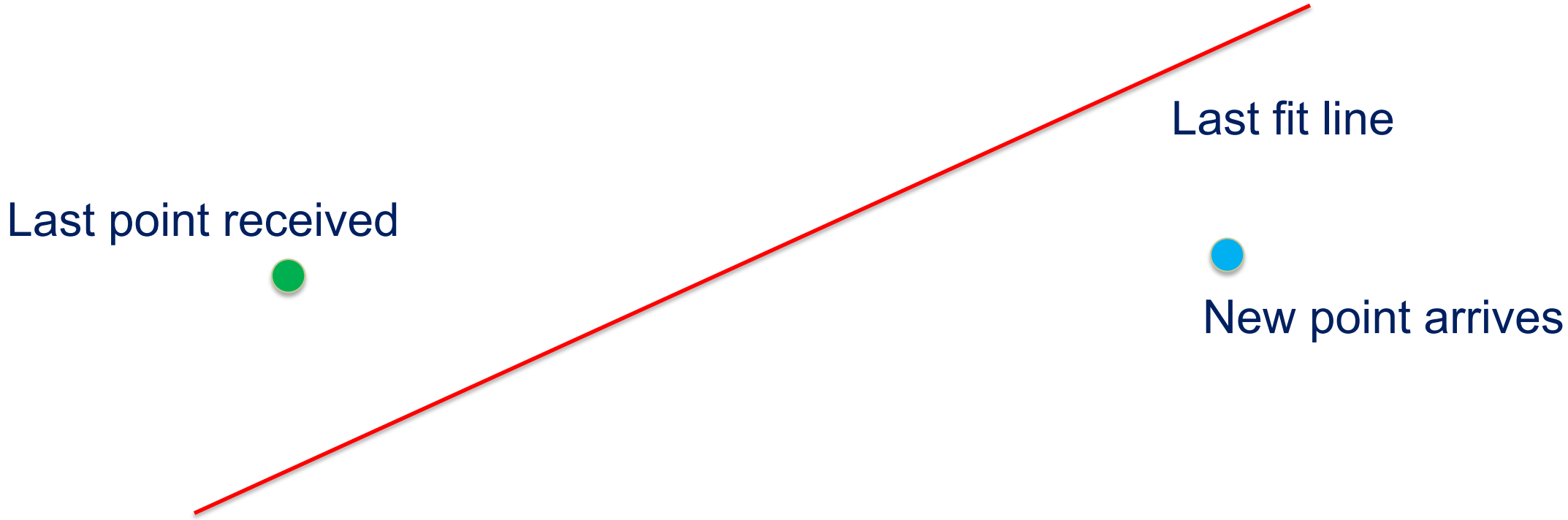
Linear Regression



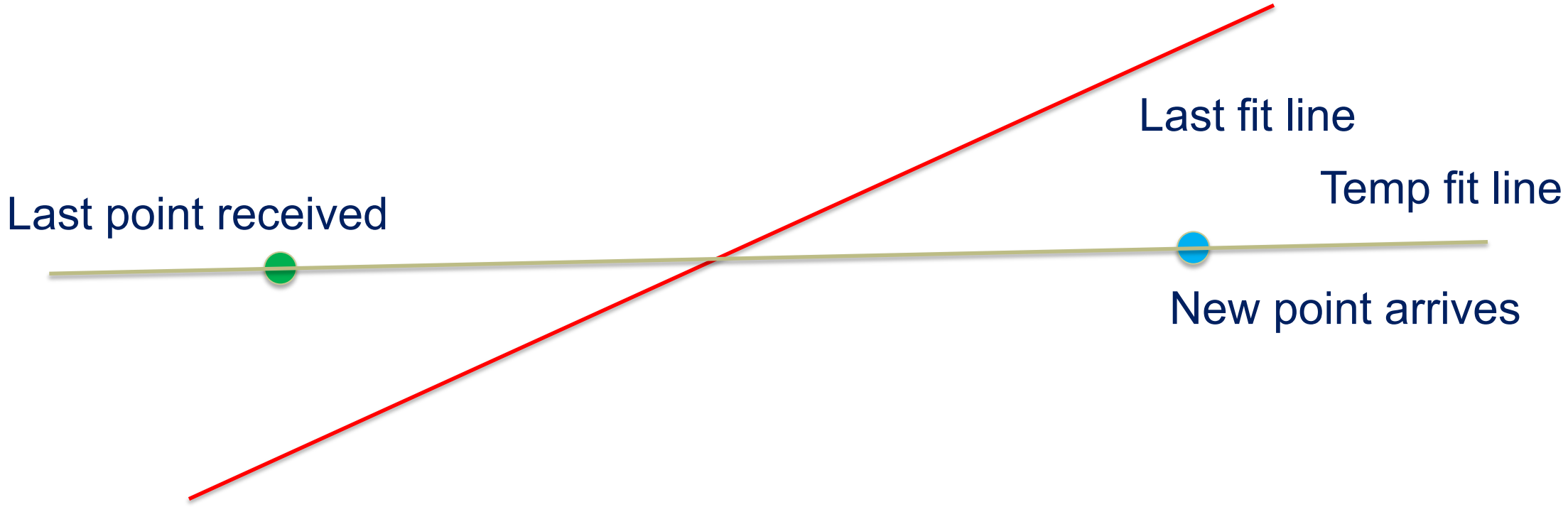
Online Linear Regression



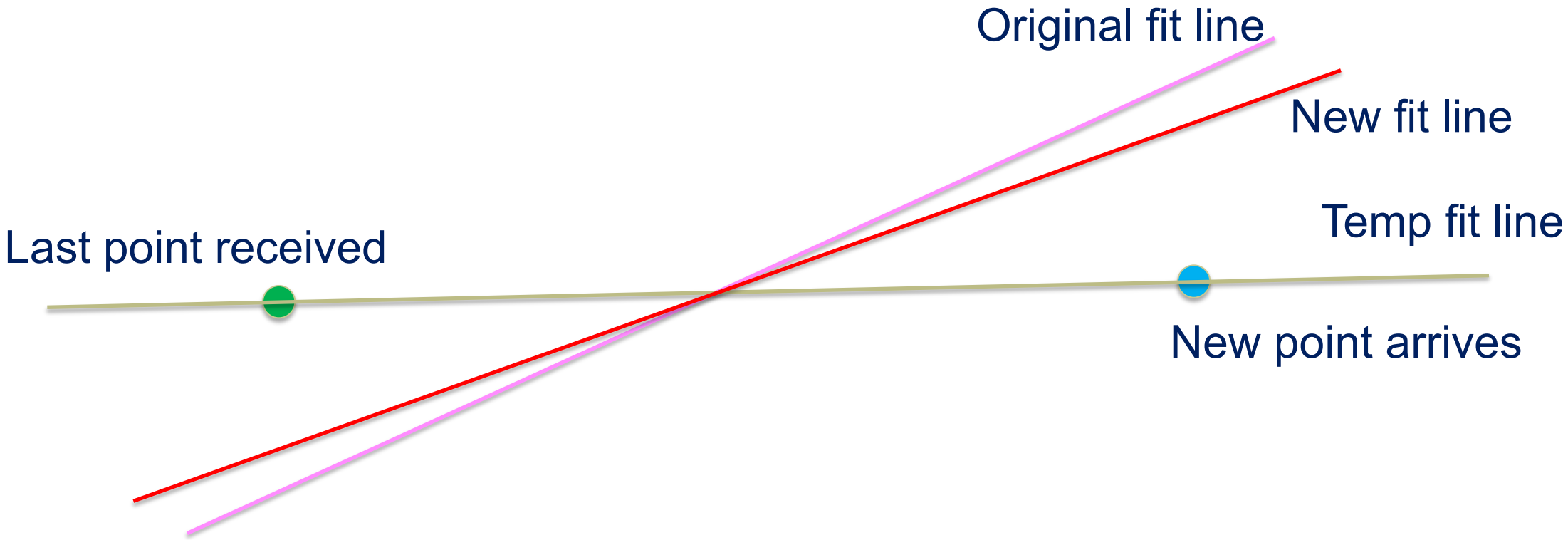
Online Linear Regression



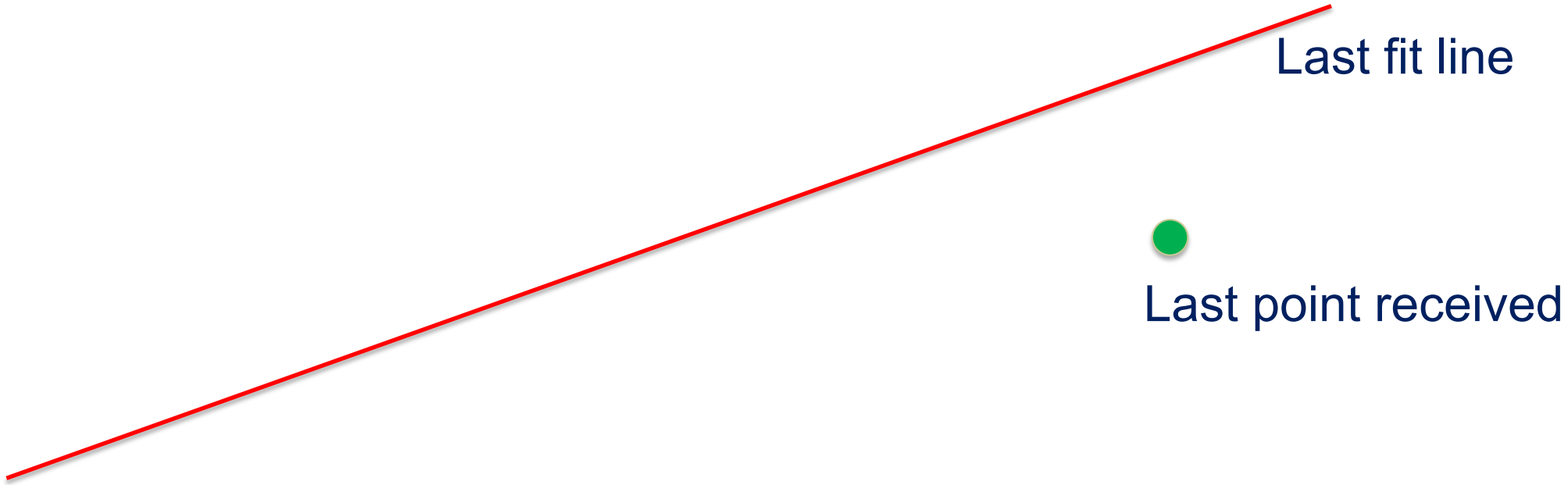
Online Linear Regression



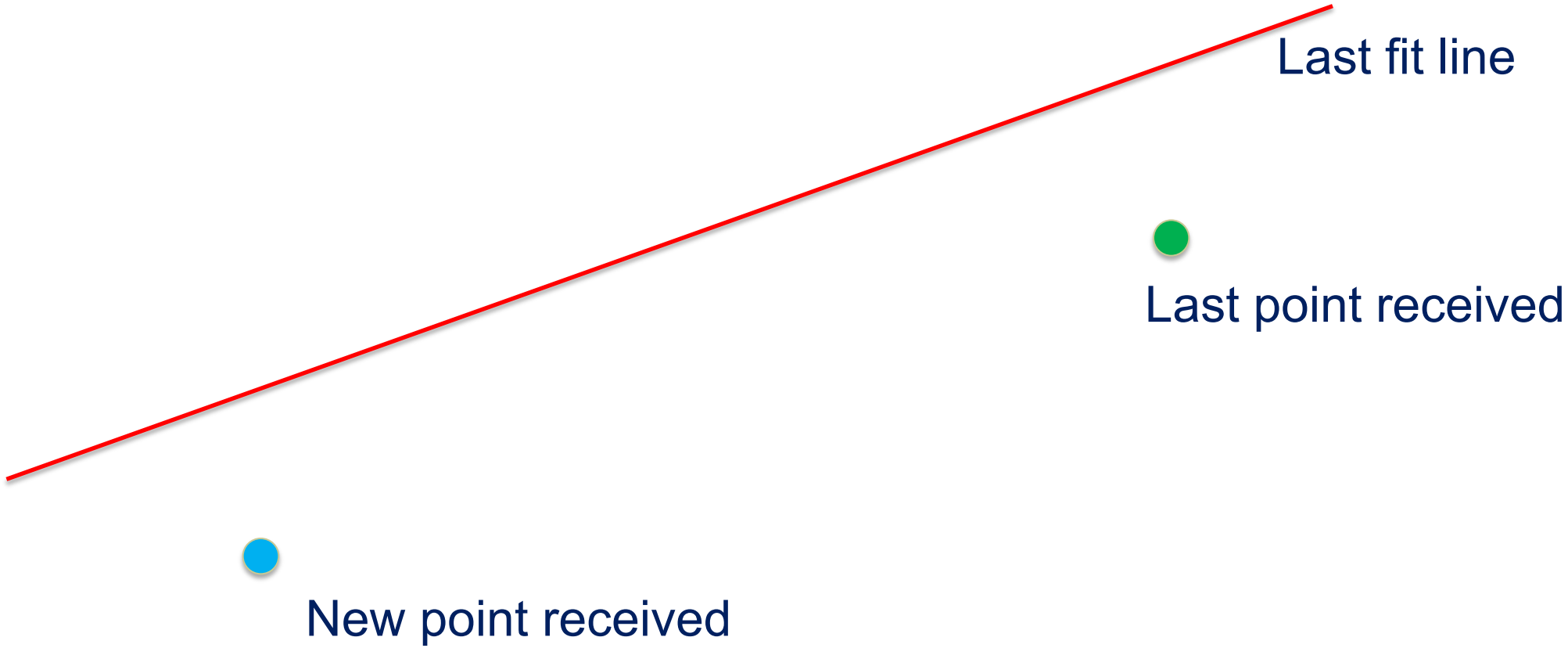
Online Linear Regression



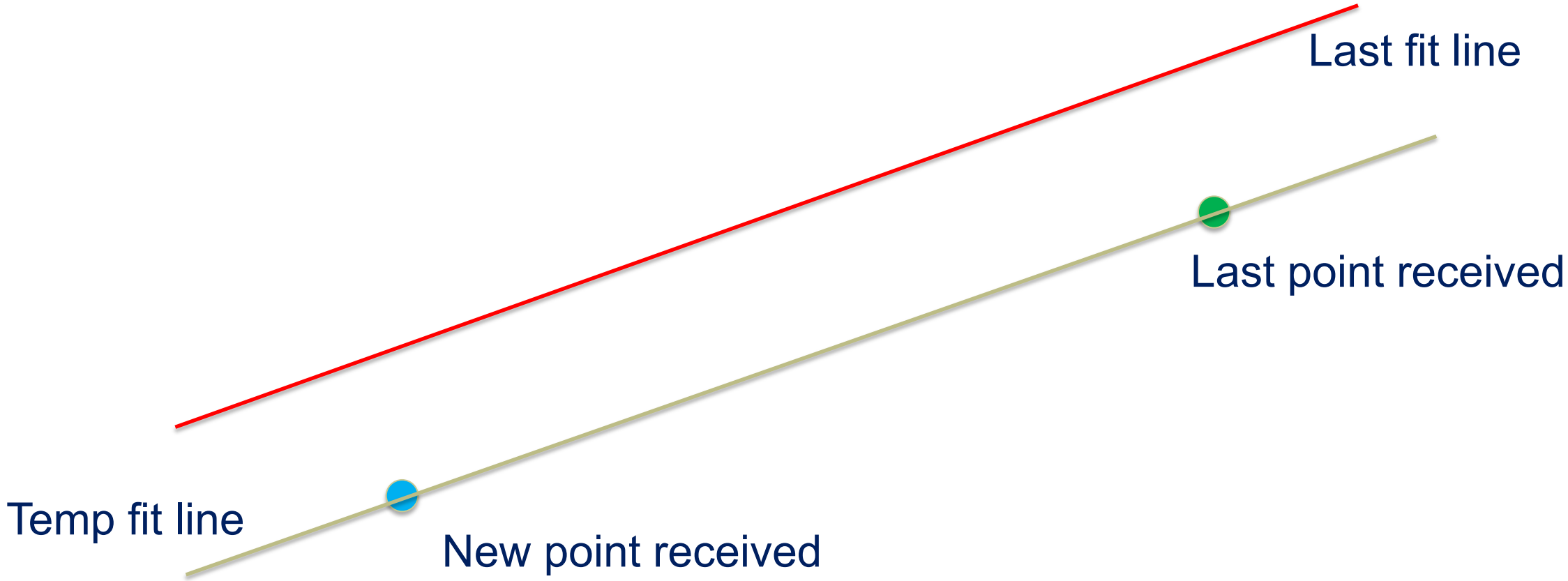
Online Linear Regression



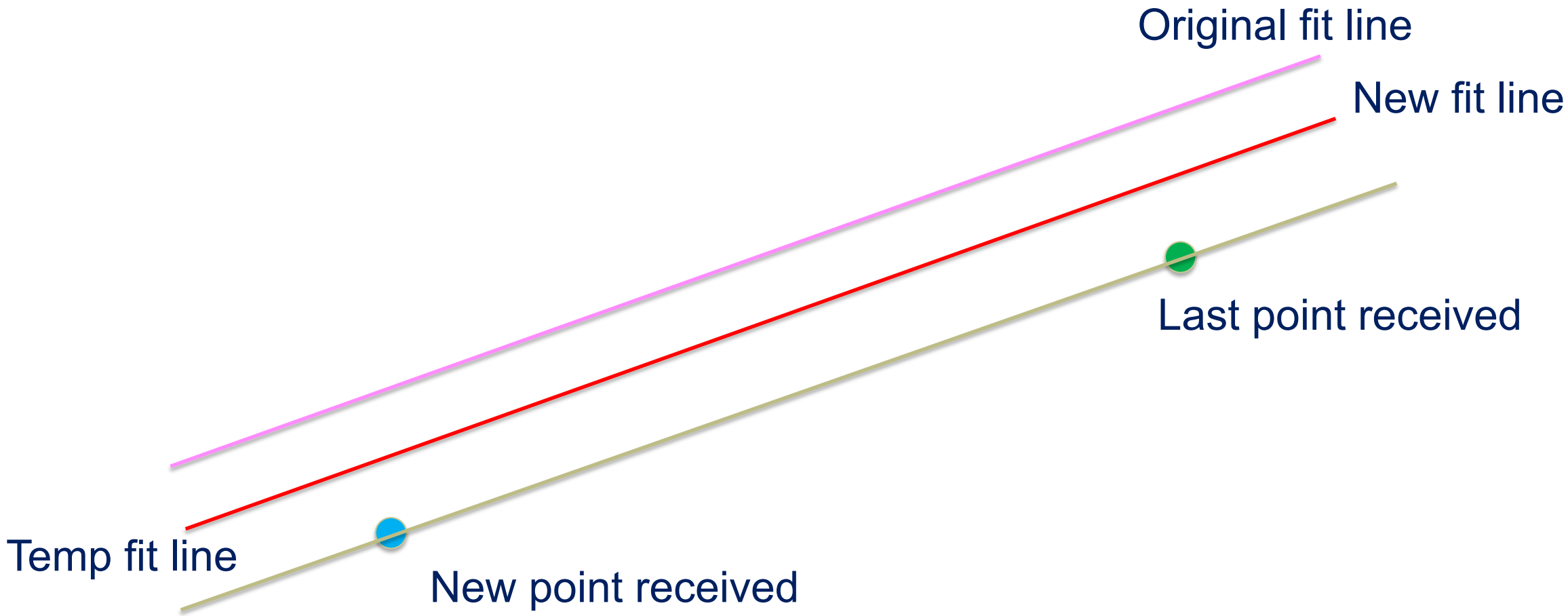
Online Linear Regression



Online Linear Regression



Online Linear Regression



Online learning – key concepts

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 $\mathrm{y}^{\mathrm{k}} \in Y$ **supervision**
4. suffer loss $L(y^k, \hat{y}^k)$ **one-pass learning (no memory)**
5. update the model for generating predictions **incremental learning** $(f^k = f^{k-1}.train())$

Online learning – key concepts

A key objective of online learning algorithms is to minimize the **regret**.

The **regret** is the difference between the performance of the online algorithm and an ideal algorithm that has been able to train on the whole data seen so far, in batch fashion; i.e., an online machine learning algorithm is trying to perform as closely as possible to an ideal corresponding offline algorithm.

Application Examples

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

Online learning – desired properties

- Learning new knowledge
- Preserving previous knowledge
- High performance
- Fast operation
- Fixed storage

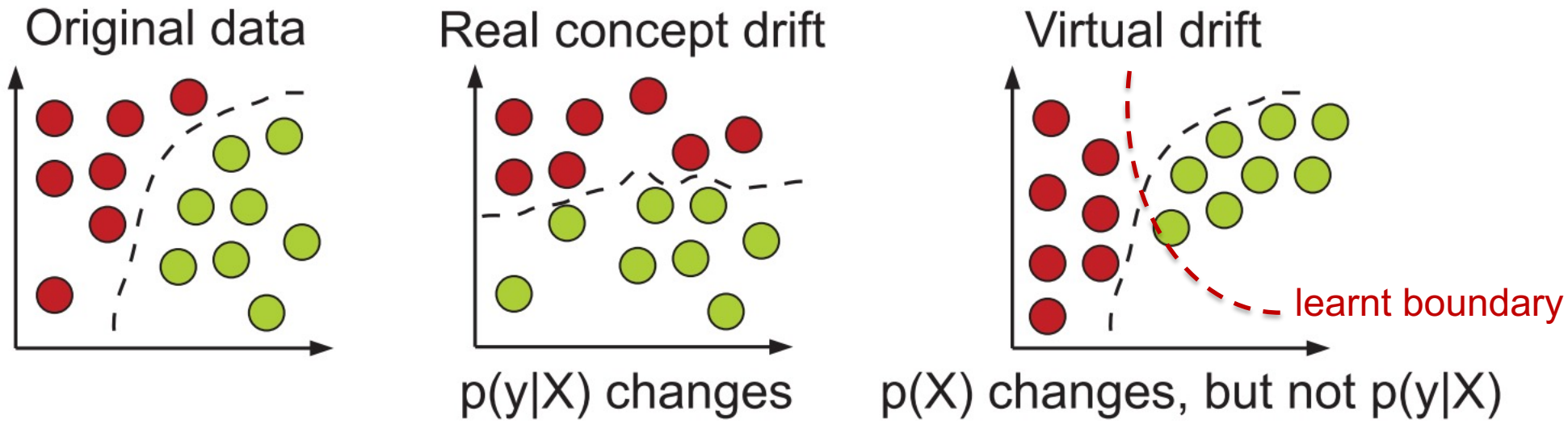
Stationary / Time-varying targets

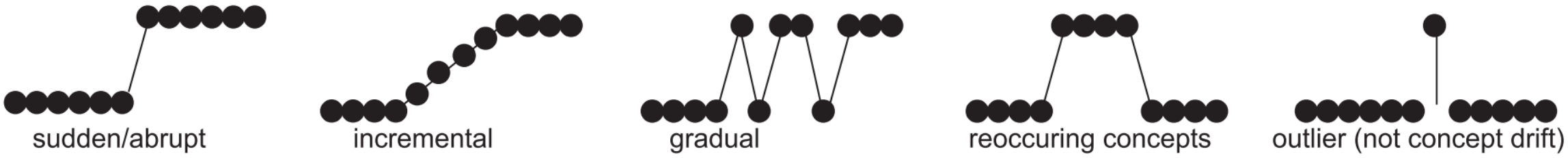
The target function that we are trying to learn maybe stationary or dynamic.

- **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).

Stationary / Time-varying targets (2)

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





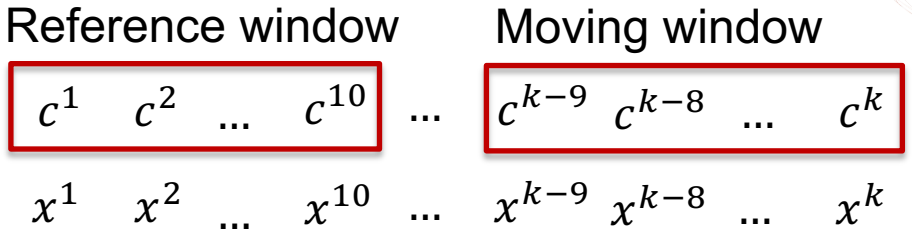
Stationary / Time-varying targets (3)

- Addressing concept drift:
  - Concept drift detection
  - Memory-based methods
  - Ensemble methods

Active methods (explicit detection)

Passive methods (implicit detection)

Concept drift detection



- **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

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

Concept drift detection (2)

- **Change detection-based**
  - Use statistical tests (e.g., binomial distribution hypothesis testing) to detect change.

- **Note**:
  - Upon drift detection, the learning model is, typically, discarded and a new one is re-trained (unlike incremental learning).

Memory-based methods

- 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

Ensemble methods

- **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

Class imbalance

- **Resampling methods**
  - Undersampling of the majority class
    - Random undersampling
  - Oversampling of the minority class
    - SMOTE[1,2] $x_{new}^{i} = x^{i} + \delta\left(x_{neigh}^{i} - x^{i}\right), \delta \in [0, 1]$
  - Data augmentation

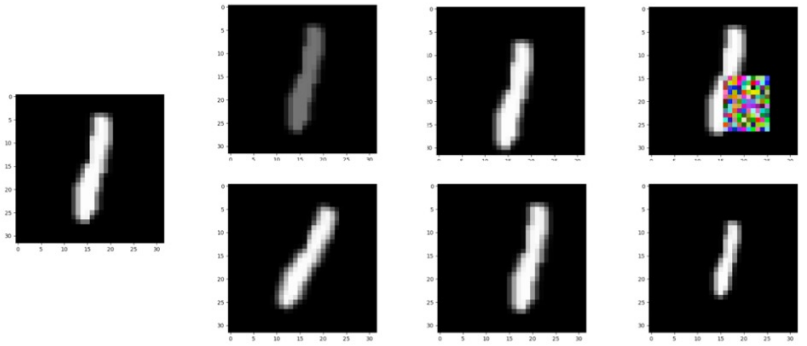


Fig. 1: Examples of augmentation transformations on the original image on the left. Top row: brightness change, height shift, random erase. Bottom row: rotation, width shift, zoom.

- **Cost-sensitive learning**
  - Assign a higher misclassification cost to minority examples

$$w_k \, L(y^k, \hat{y}^k)$$

[1] Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." *Journal of artificial intelligence research* 16 (2002): 321-357.
[2] Fernández, Alberto, et al. "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary." *Journal of artificial intelligence research* 61 (2018): 863-905.

Online leaning – other challenges

- **Catastrophic forgetting**
  - It is defined as the problem of performance degradation as new information arrives.
- **Limited storage**
  - It is unrealistic to expect that all acquired data will be available at all times.
- **Lack of supervision**
  - Unsupervised learning, semi-supervised learning, active 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[3,4,5] have been realised through AL:

Labelling malicious ads

Google Ads

Autonomous vehicles with self-driving capabilities

TESLA

[3] Sculley, D., et al. "Detecting adversarial advertisements in the wild." *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2011.
[4] NVIDIA-AI. Scalable active learning for autonomous driving. https://medium.com/nvidia-ai/scalable-active-learning-for-autonomous-driving-a-practical-implementation-and-a-b-test-4d315ed04b5f
[5] A. Karpathy. Artificial intelligence for full self-driving. https://www.youtube.com/watch?v=hx7BXih7zx8

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 AND strategy returns True:
    4. receive true answer $\mathrm{y}^{\mathrm{k}} \in Y$
    5. suffer loss $L(y^k, \hat{y}^k)$
    6. update the model for generating predictions

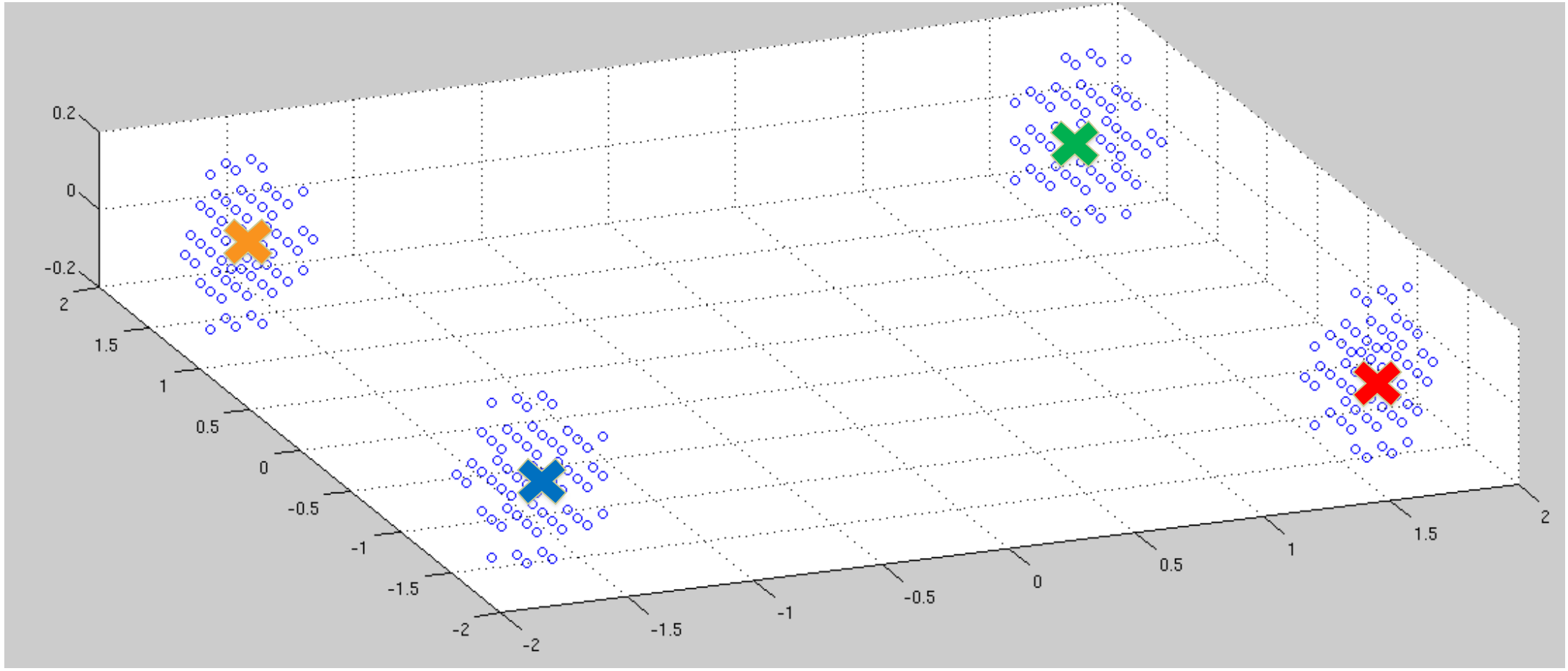
Active learning

Zero verification latency

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

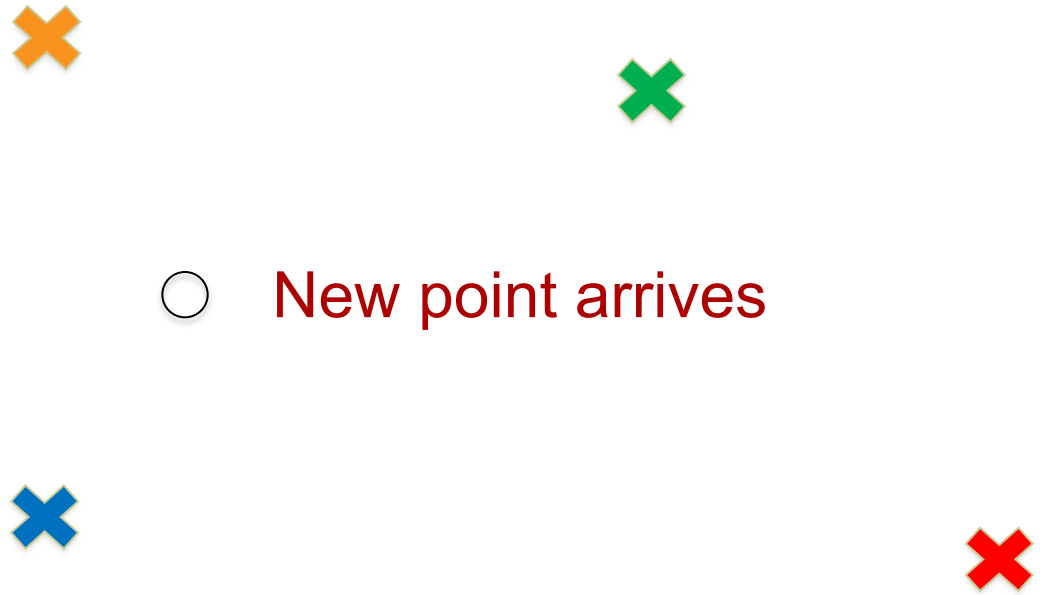


Online Clustering

Step k-1

Online Clustering

Step k



Online Clustering

Step k

Online Clustering

Step k