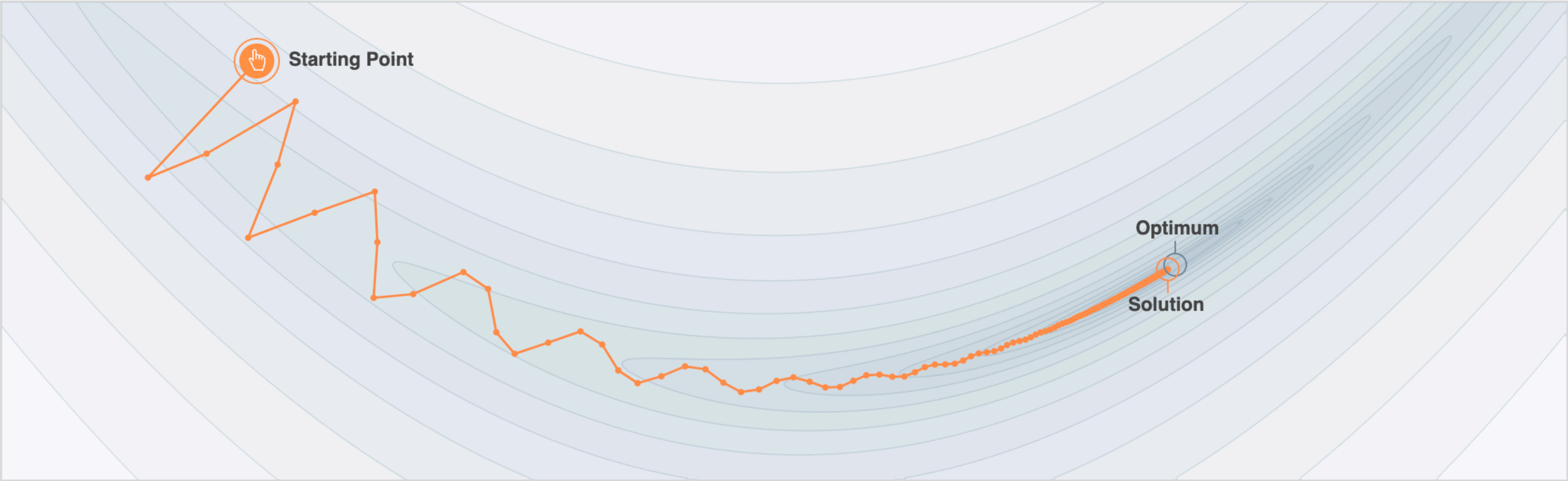
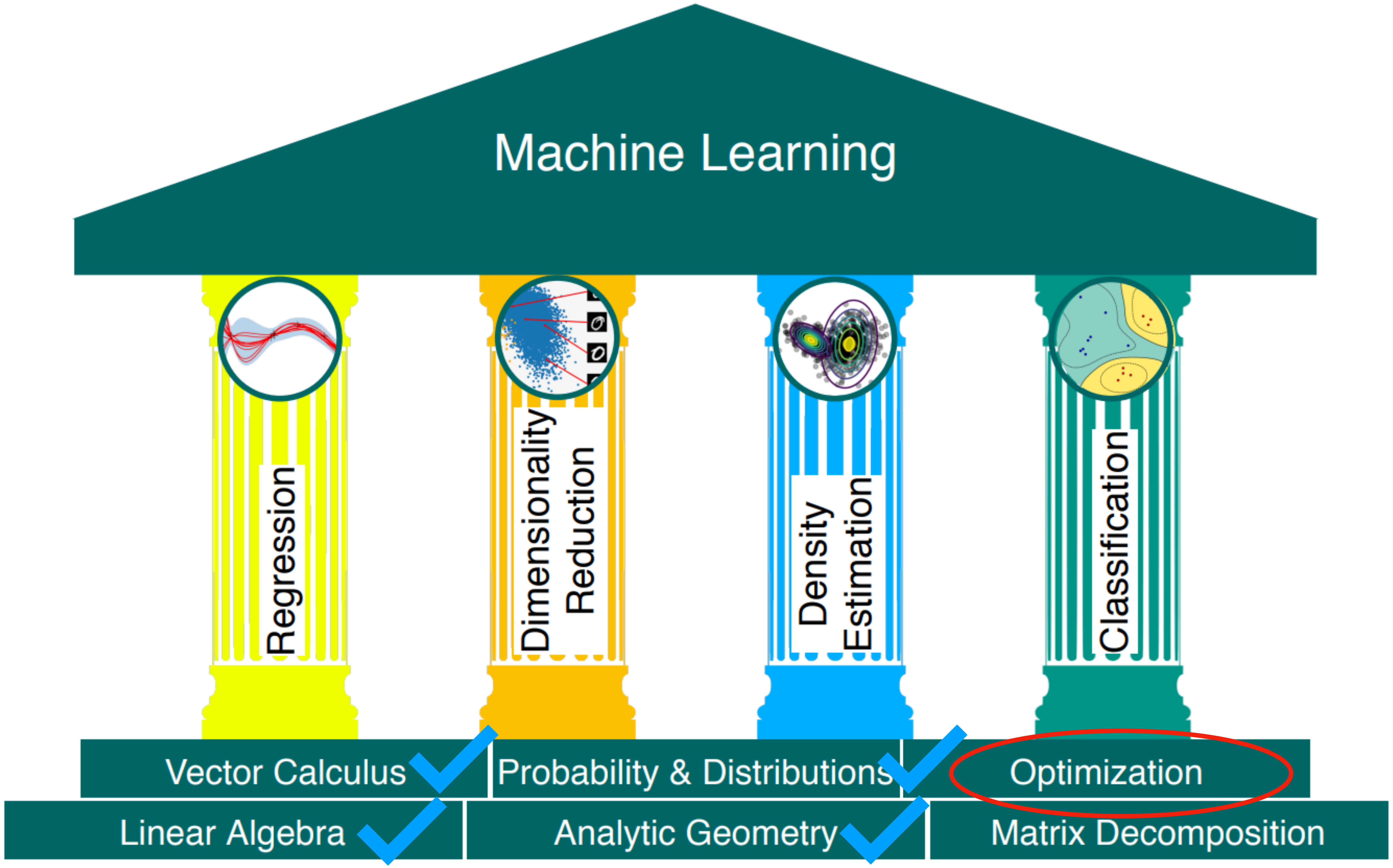
Optimisation



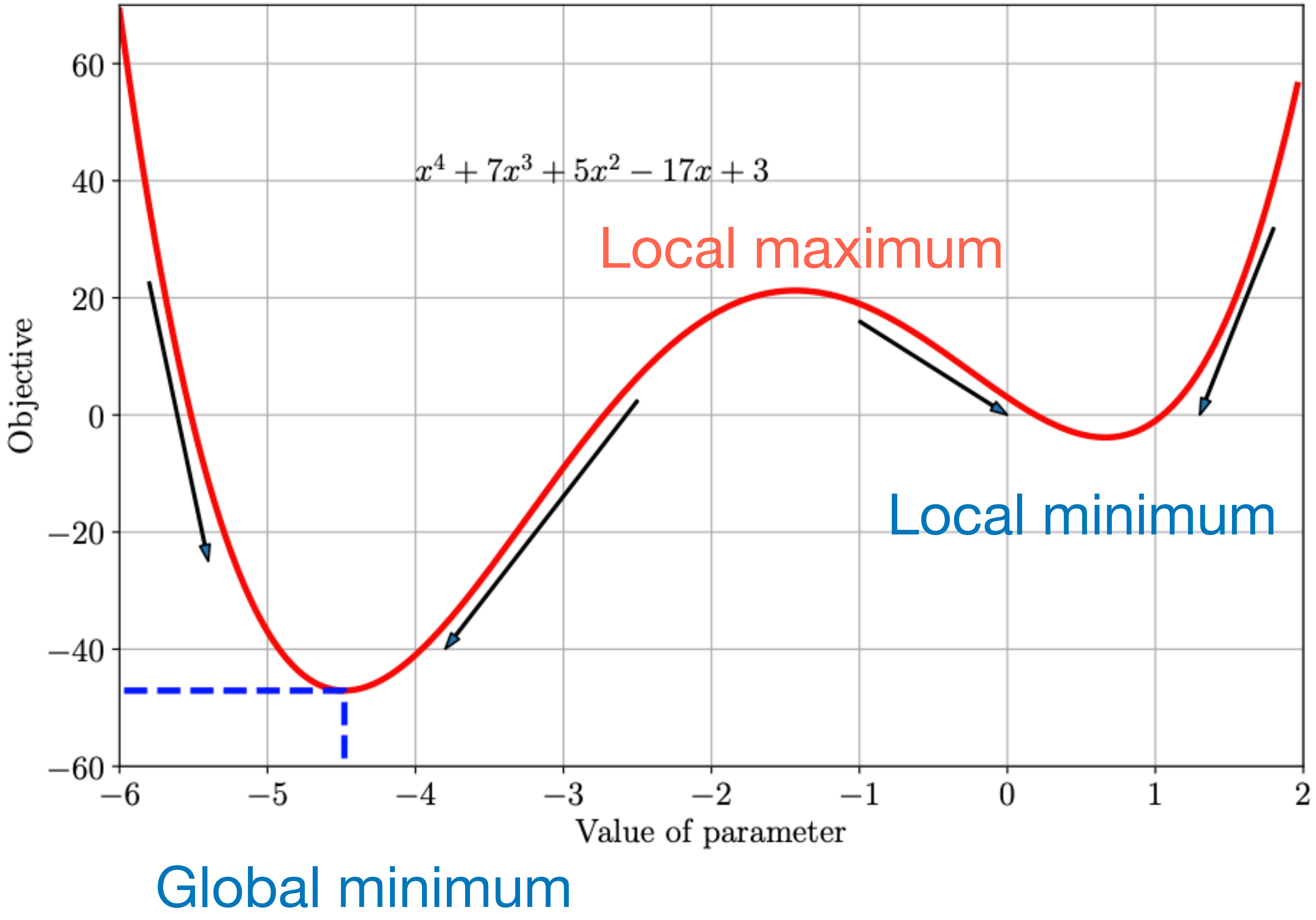
Week 7 - Introduction to ML / Thang Bui / ANU / 2023 S2

Foundations of ML



Source: Maths for ML Textbook

An analytic example



Optimisation using Gradient Descent

- Given $f : \mathbb{R}^D \to \mathbb{R}$, we consider the problem of solving for the minimum of f, $\min f$
- **Gradient descent** is a *first-order* optimisation algorithm.
- To find a local minimum of a function using gradient descent, one takes steps proportional to the **negative of the gradient** of the function at the current point.
- Gradient descent exploits the fact that $f(x_0)$ decreases **fastest** if one moves from x_0 in the direction of the negative gradient $(-\nabla f(x_0))^\intercal$ of f at x_0.
- If $x_1 = x_0 - \gamma(\nabla f(x_0))^\intercal$, for a small step size $\gamma \geq 0$, then $f(x_1) \leq f(x_0)$
- Example: https://distill.pub/2017/momentum/

Gradient descent - step size heuristic

Choosing a good step-size (learning rate) is important in gradient descent

- If the step-size is too small, gradient descent can be slow
- If the step-size is chosen too large, gradient descent can overshoot, fail to converge, or even diverge

There are several heuristics to adapt the step size

- When the function value increases after a gradient step, the step-size was too large. Undo the step and decrease the step-size
- When the function value decreases the step could have been larger. Try to increase the step-size.
- Heuristically, we choose a learning rate that starts big and ends small, e.g., $\gamma_i = 1/(i+1)$

Gradient descent with momentum

- The convergence of gradient descent may be very slow if the curvature of the optimization surface is such that there are regions that are poorly scaled
- The proposed method to improve convergence is to give gradient descent some memory
- Gradient descent with momentum is a method that introduces an additional term to remember what happened in the previous iteration.
- This memory dampens oscillations and smooths out the gradient updates
- The idea is to have a gradient update with memory to implement a moving average

$$x_{i+1} = x_i - \gamma_i m_i$$
$$m_i = (1 - \alpha) m_{i-1} + \alpha (\nabla f(x_i))^{\intercal}, \alpha \in [0,1]$$

(Stochastic) Gradient Descent for linear regression

See handwritten notes