

# MSc - Data Mining

## Topic 01 : Module Overview

---

### Part 05 : Optimisation Overview

Dr Bernard Butler and Dr Kieran Murphy

Department of Computing and Mathematics, WIT.  
([bbutler@tssg.org](mailto:bbutler@tssg.org) and [kmurphy@wit.ie](mailto:kmurphy@wit.ie))

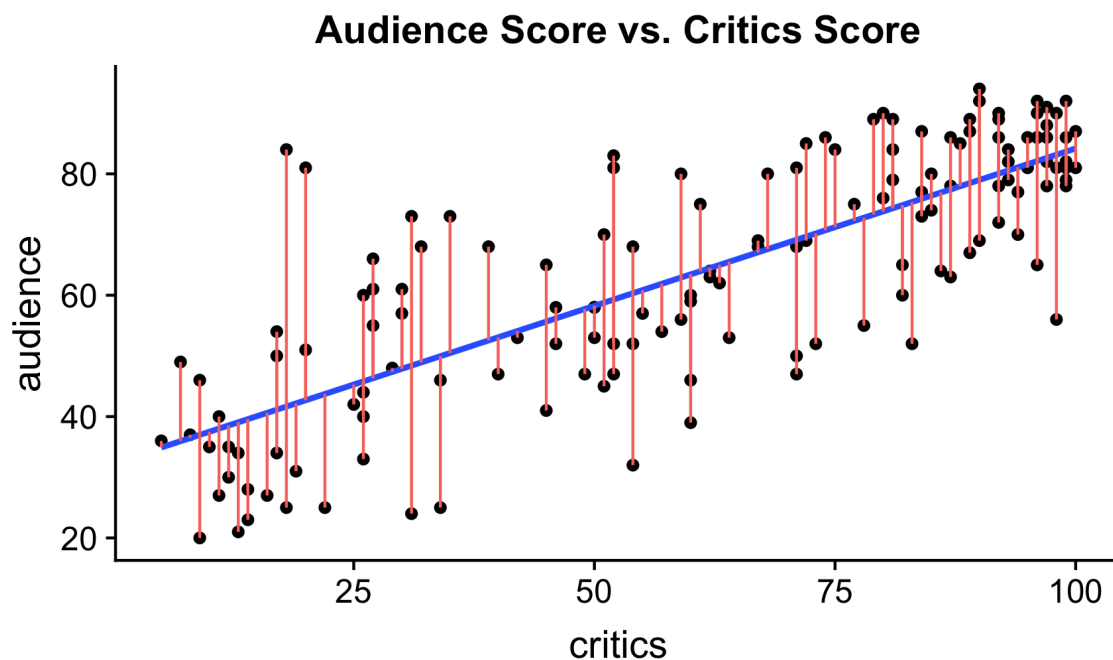
Spring Semester, 2021

Outline

# Machine learning meta-model: The Loss Function

- Machine learning is a large part of this module, but how does it actually work?
- Mathematically, we have a *function*, of one or more variables.
- Most machine learning problems can be formulated as finding values of that function that satisfy certain desirable properties
- Often that function is referred to as a *loss function*  $L \equiv L(M(\{D_i\}, \mathbf{a}), \{\varepsilon_i\})$ , where
  - $\{D_i\}$ , with  $i = 1, \dots, m$  represents the *training* data (observations) used by the learner;
  - $\{\varepsilon_i\}$  represents the (unknown) errors in that training data;
  - $M(\{D_i\}, \mathbf{a})$  represents the model used to represent the data;
  - $\mathbf{a}$  represents one or more variables, that each take a special value when the required property holds
- Generally, the property we are looking for is that the value of the Loss Function should be as small as possible.

## Example Loss Function



Here the training data is  $\{D_i\} = \{x_i, y_i\}$  where  $x_i$  is the  $i^{\text{th}}$  critics score and  $y_i$  is the corresponding audience score. A linear relationship  $M : y^* = a_0 + a_1x$  is assumed and the errors are estimated by the difference between the predicted values (on the line) and the corresponding data values.

*Source:*

*towardsdatascience.com*

- The loss function is an expression computed from all the error estimates, giving a scalar output (a single number) with the property that the loss function decreases when the overall error decreases.
- So: minimising the loss function has the effect of fitting the line as close to the data as possible, equivalent to searching for the “best” values of  $a_0$  and  $a_1$  above.
- Many machine learning algorithms can be formulated in this way.

# Solving the optimisation problem

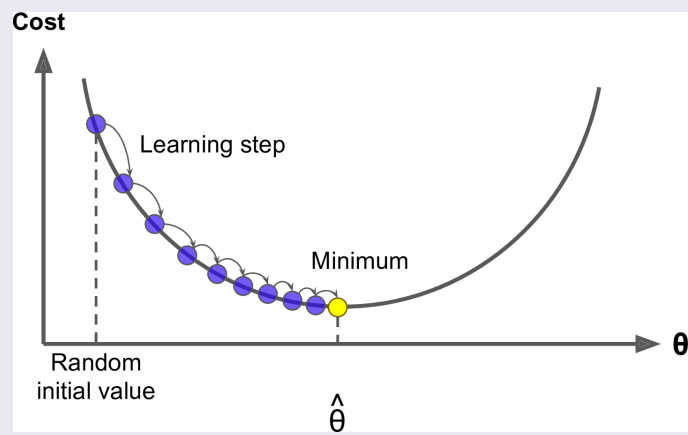
---

- ① Use trial and error - unworkable unless there is a small, finite set to check
- ② Use function values only, compare them and use heuristics to guide the search
- ③ Use derivatives and head downhill until you reach a valley (gradient descent)
- ④ Use higher order derivatives to make more informed decisions

*Enhancement:* Apply constraints, e.g., when predicting weight, it cannot take negative values!

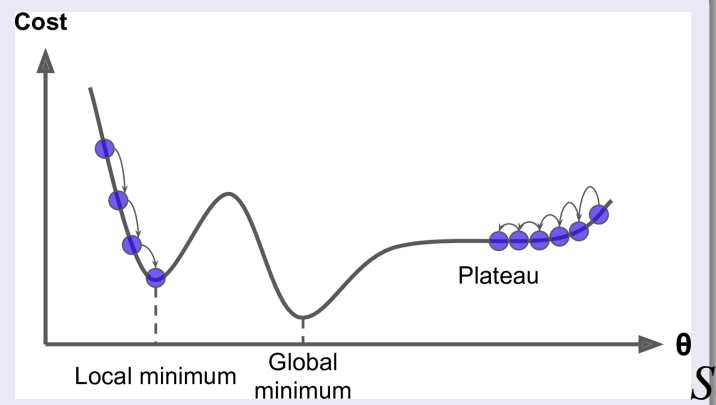
# Spotlight on Gradient Descent

## Basic Operation



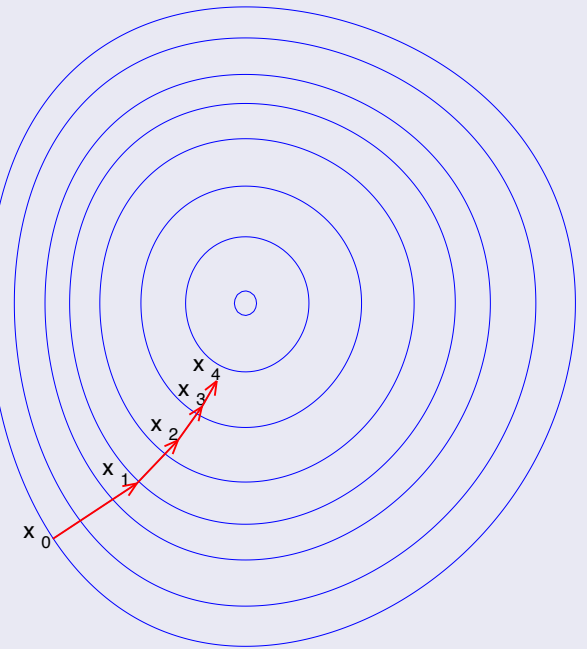
*Hands-on machine learning*

## Challenges



*Hands-on machine learning*

## Two dimensions - Contours



*Wikipedia*