Regularisation

Procheta Sen

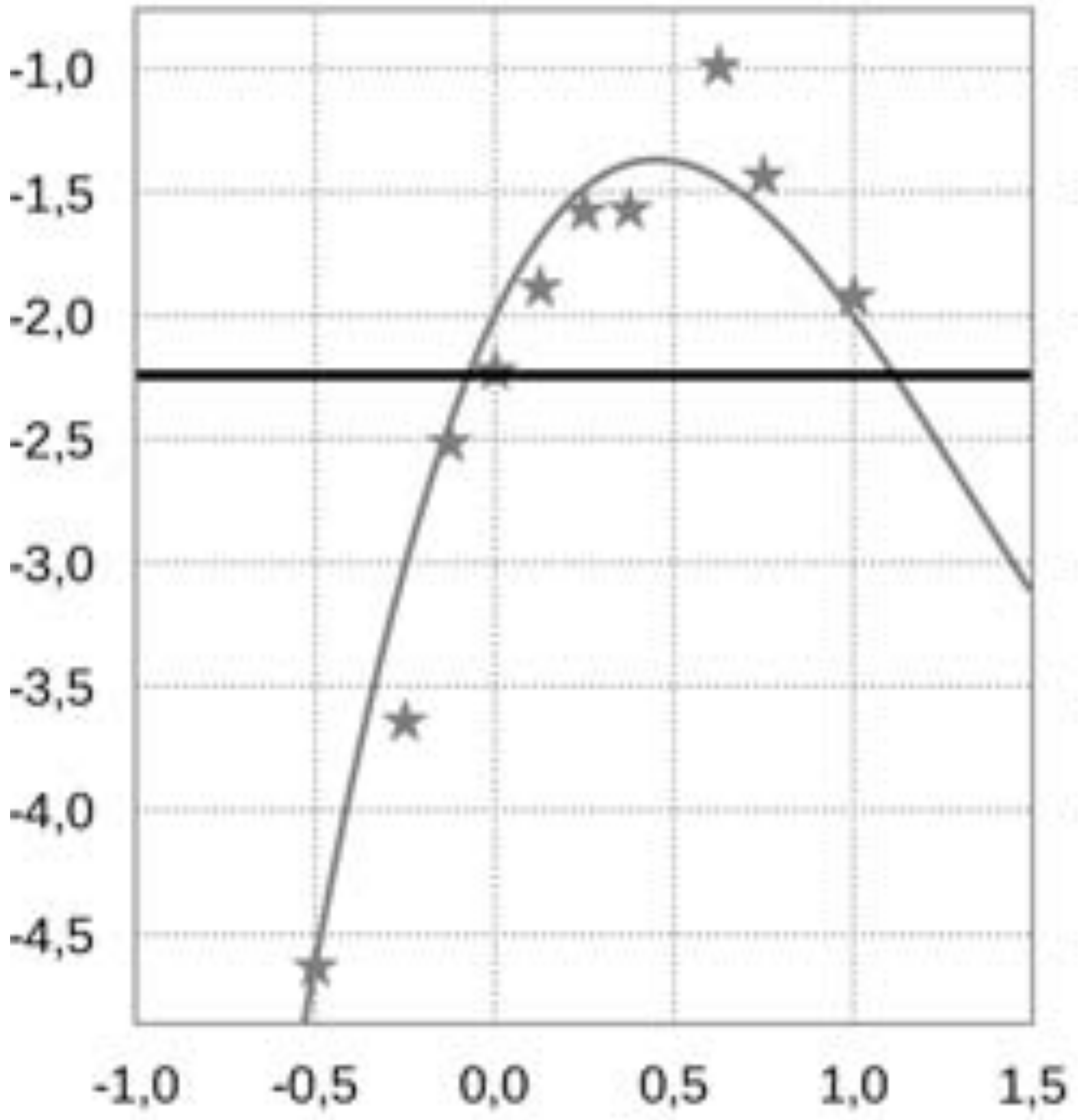
UNIVERSITY OF LIVERPOOL

Regularisation

- **Regularisation** is a process of reducing overfitting in a model by constraining it (reducing the complexity/no. of parameters)
- For classifiers that use a weight vector, regularisation can be done by minimising the norm (length) of the weight vector.
- Several popular regularisation methods exist
  - L2 regularisation (ridge regression or Tikhonov regularisation)
  - L1 regularisation (Lasso regression)
  - L1+L2 regularisation (mixed regularisation)

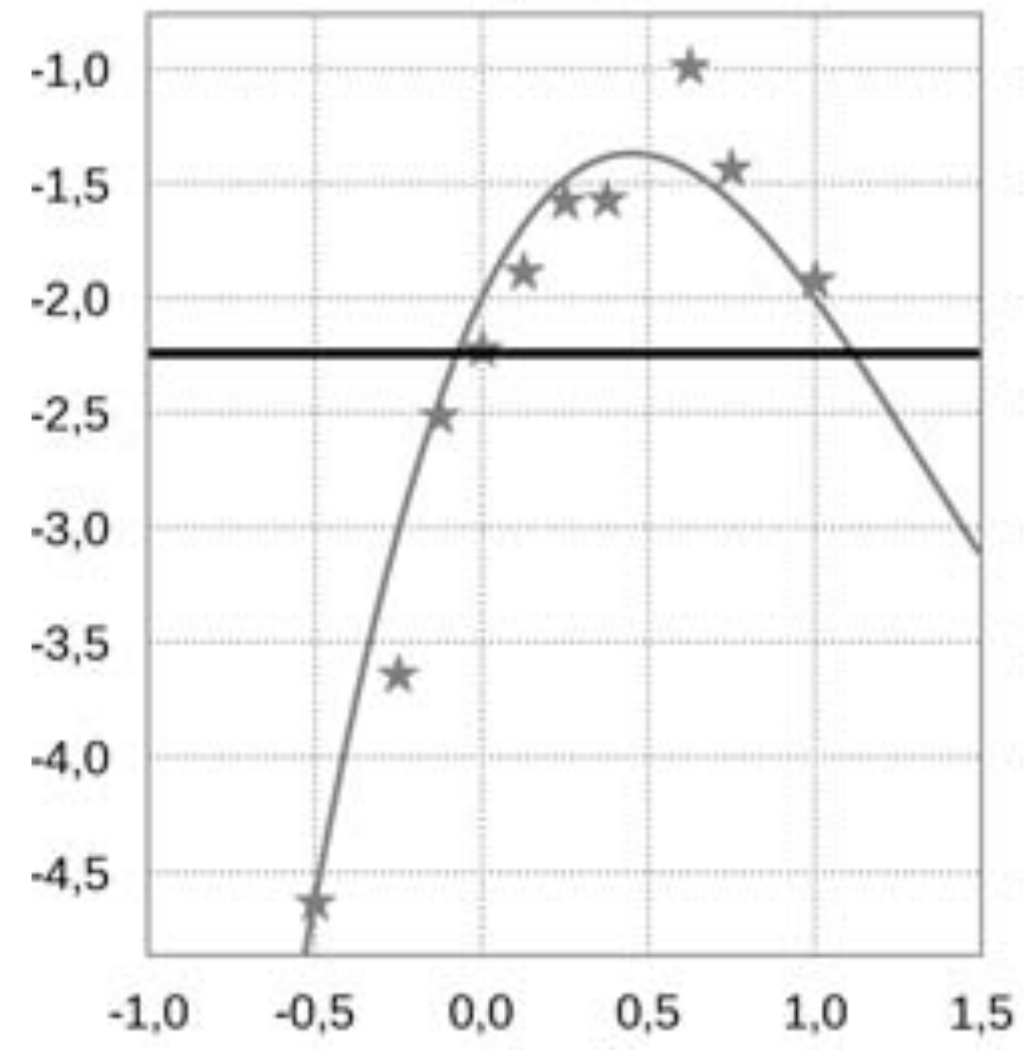
Approximation of $f(x) = x^3 - 4x^2 + 3x - 2$ by polynomials of degree d

The dataset $\mathscr{D} = \{(x_i, y_i)\}_{i=1}^n$ is sampled from the polynomial $f(x) = x^3 - 4x^2 + 3x - 2$ with some noise



Approximation of $f(x) = x^3 - 4x^2 + 3x - 2$ by polynomials of degree d

The dataset $\mathscr{D} = \{(x_i, y_i)\}_{i=1}^{n}$ is sampled from the polynomial $f(x) = x^3 - 4x^2 + 3x - 2$ with some noise



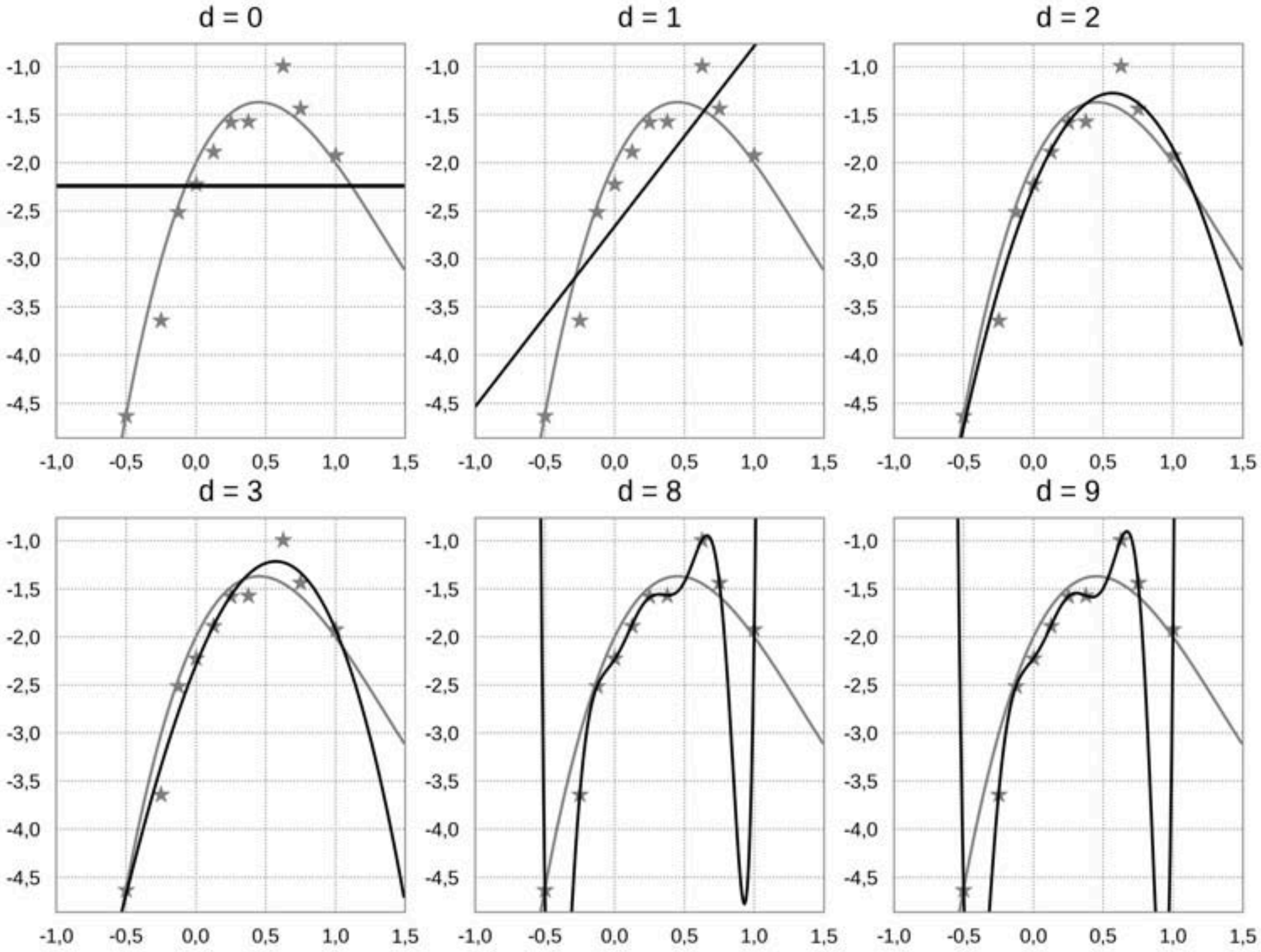
We want to approximate the **unknown** function $f(x)$ by a polynomial of degree d

$$\hat{y}(x, \overline{W}) = w_0 + \sum_{j=1}^{d} w_j x^j = (1, x, x^2, \ldots, x^d) \cdot \overline{W}$$

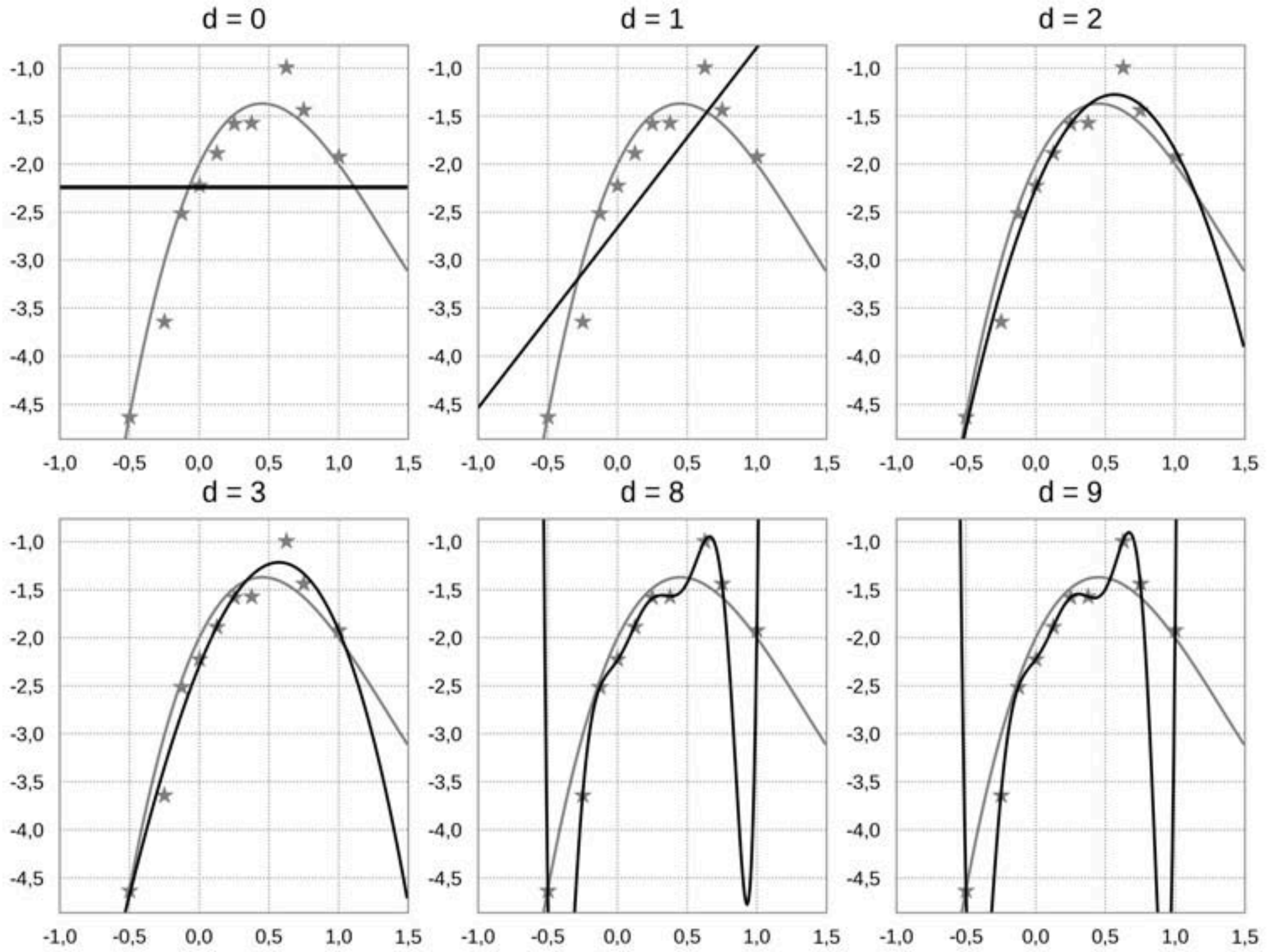
such that the following loss function (residual sum of squares (RSS)) is minimised

$$L(\mathscr{D}, \overline{W}) = \sum_{i=1}^{n} (\hat{y}(x_i, \overline{W}) - y_i)^2$$

Approximation of $f(x) = x^3 - 4x^2 + 3x - 2$ by polynomials of degree d



Approximation of $f(x) = x^3 - 4x^2 + 3x - 2$ by polynomials of degree d

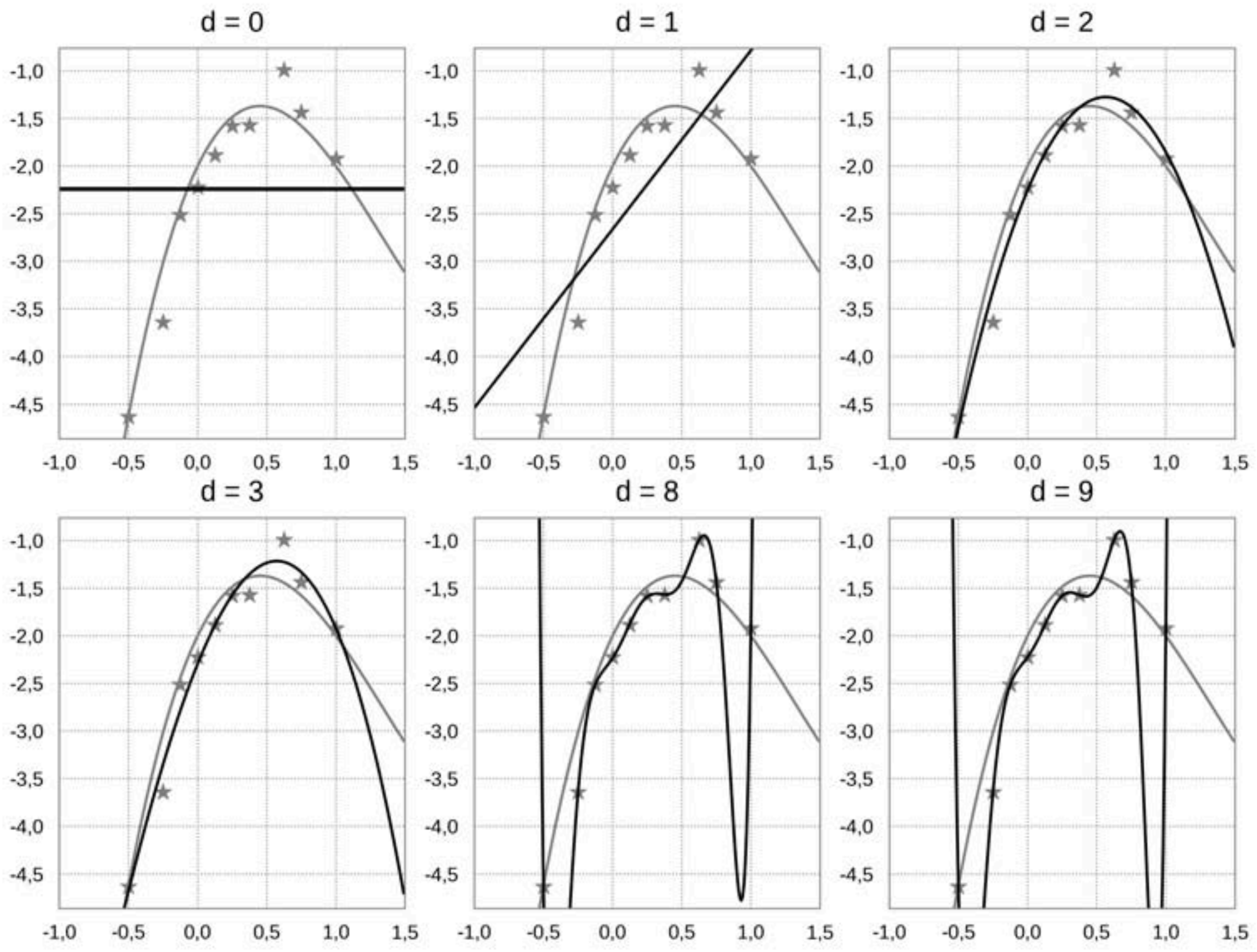


$$f_0(x) = -2{,}2393,$$
$$f_1(x) = -2{,}6617 + 1{,}8775x,$$
$$f_2(x) = -2{,}2528 + 3{,}4604x - 3{,}0603x^2,$$
$$f_3(x) = -2{,}2937 + 3{,}5898x - 2{,}6538x^2 - 0{,}5639x^3,$$
$$f_8(x) = -2{,}2324 + 2{,}2326x + 6{,}2543x^2 + 15{,}5996x^3 - 239{,}9751x^4 + \\ + 322{,}8516x^5 + 621{,}0952x^6 - 1478{,}6505x^7 + 750{,}9032x^8,$$
$$f_9(x) = -2{,}22 + 2{,}01x + 4{,}88x^2 + 31{,}13x^3 - 230{,}31x^4 + \\ + 103{,}72x^5 + 869{,}22x^6 - 966{,}67x^7 - 319{,}31x^8 + 505{,}64x^9.$$

The parameters grow!

Let’s try to restrict their growth

Approximation of $f(x) = x^3 - 4x^2 + 3x - 2$ by polynomials of degree d



$$f_0(x) = -2{,}2393,$$

$$f_1(x) = -2{,}6617 + 1{,}8775x,$$

$$f_2(x) = -2{,}2528 + 3{,}4604x - 3{,}0603x^2,$$

$$f_3(x) = -2{,}2937 + 3{,}5898x - 2{,}6538x^2 - 0{,}5639x^3,$$

$$f_8(x) = -2{,}2324 + 2{,}2326x + 6{,}2543x^2 + 15{,}5996x^3 - 239{,}9751x^4 + \\ + 322{,}8516x^5 + 621{,}0952x^6 - 1478{,}6505x^7 + 750{,}9032x^8,$$

$$f_9(x) = -2{,}22 + 2{,}01x + 4{,}88x^2 + 31{,}13x^3 - 230{,}31x^4 + \\ + 103{,}72x^5 + 869{,}22x^6 - 966{,}67x^7 - 319{,}31x^8 + 505{,}64x^9.$$

$$L(\mathcal{D}, \overline{W}) = \sum_{i=1}^{n} (\hat{y}(x_i, \overline{W}) - y_i)^2 + \lambda ||W||_2^2$$

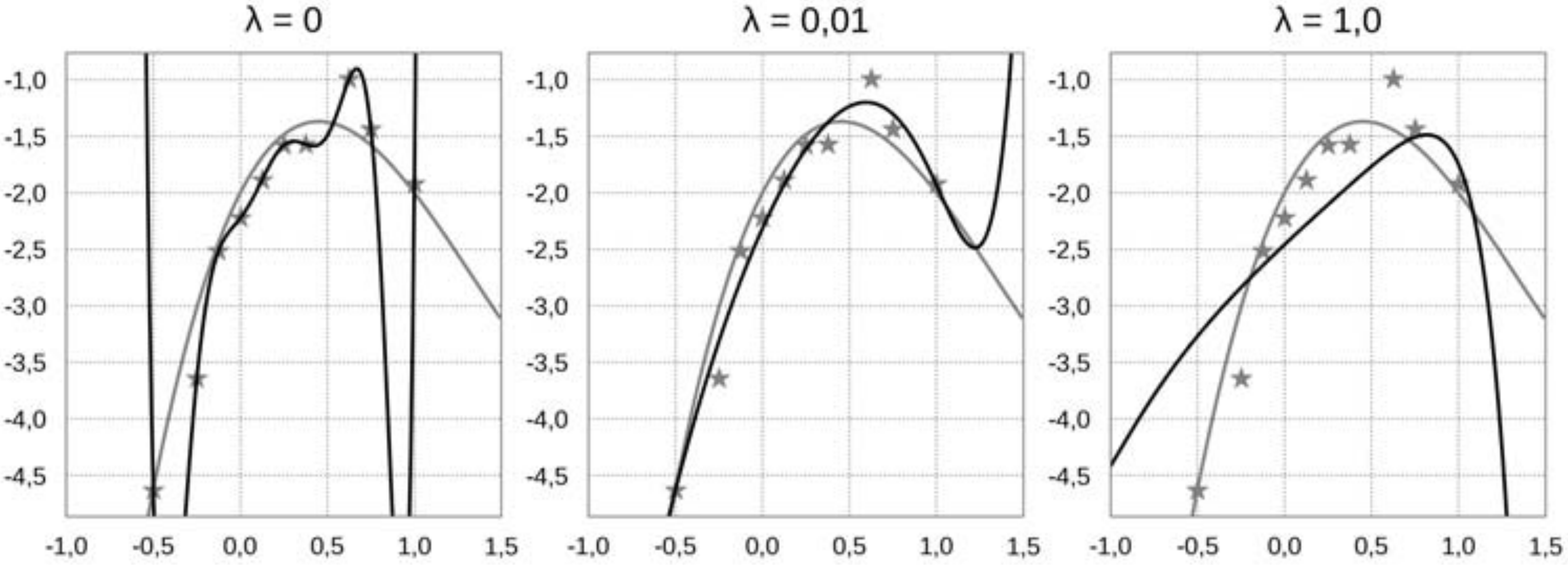
Approximation of $f(x) = x^3 - 4x^2 + 3x - 2$ by polynomials of degree d

$$L(\mathcal{D}, \overline{W}) = \sum_{i=1}^{n} (\hat{y}(x_i, \overline{W}) - y_i)^2 + \lambda ||W||_2^2$$

$$\begin{aligned}
f_{\lambda=0}(x) = &- 2{,}22 + 2{,}01x + 4{,}88x^2 + 31{,}13x^3 - 230{,}31x^4 + \\
&+ 103{,}72x^5 + 869{,}22x^6 - 966{,}67x^7 - 319{,}31x^8 + 505{,}64x^9, \\
f_{\lambda=0{,}01}(x) = &- 2{,}32 + 3{,}40x - 2{,}33x^2 + 0{,}05x^3 - 0{,}51x^4 - \\
&- 0{,}29x^5 - 0{,}22x^6 - 0{,}06x^7 + 0{,}09x^8 + 0{,}24x^9, \\
f_{\lambda=1}(x) = &- 2{,}46 + 1{,}45x - 0{,}19x^2 + 0{,}22x^3 - 0{,}13x^4 - \\
&- 0{,}05x^5 - 0{,}14x^6 - 0{,}13x^7 - 0{,}16x^8 - 0{,}16x^9.
\end{aligned}$$

Approximation of $f(x) = x^3 - 4x^2 + 3x - 2$ by polynomials of degree d

$$
\begin{aligned}
f_{\lambda=0}(x) = &-2{,}22 + 2{,}01x + 4{,}88x^2 + 31{,}13x^3 - 230{,}31x^4 + \\
&+ 103{,}72x^5 + 869{,}22x^6 - 966{,}67x^7 - 319{,}31x^8 + 505{,}64x^9, \\
f_{\lambda=0{,}01}(x) = &-2{,}32 + 3{,}40x - 2{,}33x^2 + 0{,}05x^3 - 0{,}51x^4 - \\
&- 0{,}29x^5 - 0{,}22x^6 - 0{,}06x^7 + 0{,}09x^8 + 0{,}24x^9, \\
f_{\lambda=1}(x) = &-2{,}46 + 1{,}45x - 0{,}19x^2 + 0{,}22x^3 - 0{,}13x^4 - \\
&- 0{,}05x^5 - 0{,}14x^6 - 0{,}13x^7 - 0{,}16x^8 - 0{,}16x^9.
\end{aligned}
$$



L2 regularisation

- Let us denote by $L(\mathscr{D}, \overline{W})$ the Loss of classifying the dataset $\mathscr{D}$ using the model represented by the weight vector $\overline{W}$
- We would like to impose L2 regularisation on $\overline{W}$.
- The overall objective to minimise can then be written as follows

$$J(\mathscr{D}, \overline{W}) = L(\mathscr{D}, \overline{W}) + \lambda \| \overline{W} \|_2^2 = L(\mathscr{D}, \overline{W}) + \lambda \sum_{i=1}^{d} w_i^2.$$

Here λ is called the **regularisation coefficient** and is usually set via **cross-validation**.

- The gradient of the overall objective simply becomes the sum of the loss-gradient and the scaled weight vector $\overline{W}$.

$$\nabla_{\overline{W}} J(\mathscr{D}, \overline{W}) = \nabla_{\overline{W}} L(\mathscr{D}, \overline{W}) + 2\lambda \overline{W}$$

Examples

- Note that SGD update rule for minimising a loss multiplies the loss gradient by a negative learning rate (μ).

- Therefore, the L2 regularised update rules will have a $-2\mu\lambda\overline{W}$ term as shown in the following examples

Example

L2 regularised Perceptron update rule

$$
\begin{aligned}
\overline{W} &\leftarrow \overline{W} - \mu\left(-y_i \cdot \overline{X}_i + 2\lambda\overline{W}\right) \\
&= \overline{W} + \mu \cdot y_i \cdot \overline{X}_i - 2\mu\lambda\overline{W} \\
&= \overline{W} + y_i \cdot \overline{X}_i - 2\lambda\overline{W} \qquad \text{(for } \mu = 1\text{)} \\
&= (1 - 2\lambda) \cdot \overline{W} + y_i \cdot \overline{X}_i
\end{aligned}
$$

How to set λ

- Split your training dataset into **training** and **validation** parts (e.g. 80%-20%)

- Try different values for λ (typically in the logarithmic scale), e.g.

$$\lambda = 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 0, 10^{1}, 10^{2}, 10^{3}, 10^{4}, 10^{5}$$

- Train a different classification model for each λ and select the value that gives the best performance (e.g. accuracy, RSS, etc.) on the validation data.