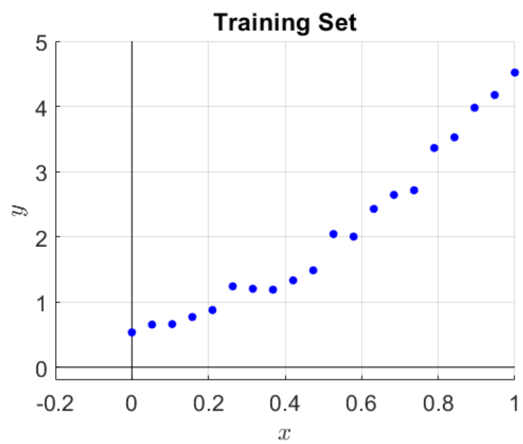
Introduction to Machine Learning
Lecture 6 - Regression

1 Introduction

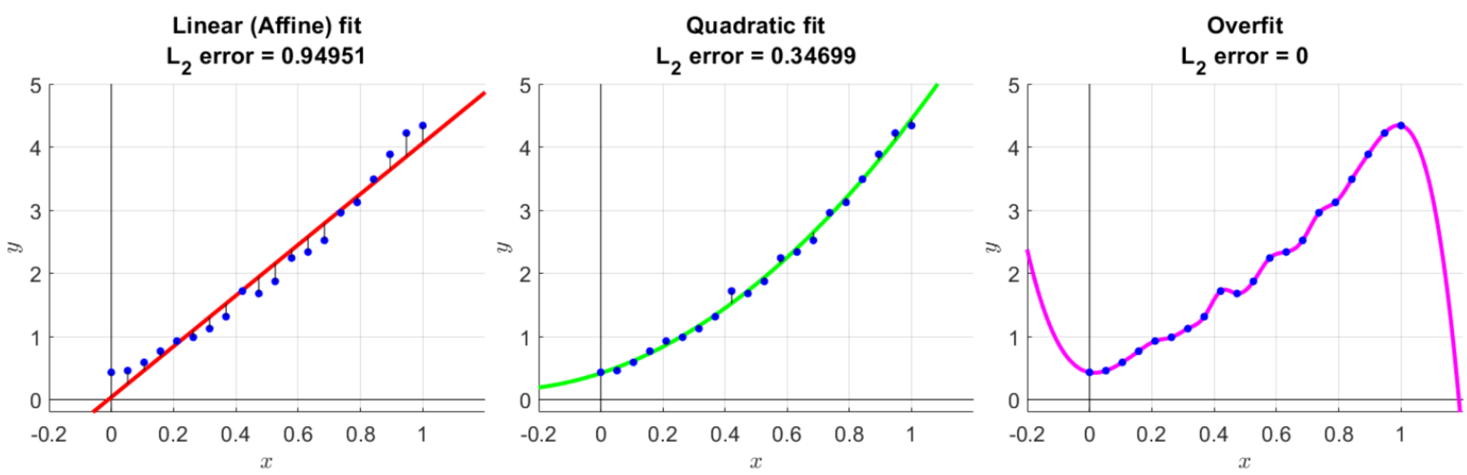
Consider a training set $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$ such that:

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i$$

where ϵ_i is assumed to be small and f is an unknown function.
For example:



In the regression problem we search $\hat{f}$ which estimates (fits) the unknown f.
Some possible options to $\hat{f}$:



The L_2 error is given by:

$$L_2\text{-error} \triangleq \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{f}(\boldsymbol{x}_i)\right)^2$$

On the left, we assume a simple model: a linear (affine) function and obtain a relatively high error.
On the right, we assume a complex model which obtains zero error.
However, the quadratic model (middle) which obtains relatively small (but not zero) error seems to be the most reasonable estimation.

2 Least Squares (Linear) Regression

2.1 1D

2.1.1 Linear fit

Let $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ where $x_i, y_i \in \mathbb{R}$ be the training set.
We assume the following model:

$$\hat{f}_{\text{Linear}}(x) = wx, \qquad w \in \mathbb{R}$$

The L_2 error (MSE loss) is given by:

$$L(w) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{f}(x_i)\right)^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - wx_i)^2 = \frac{1}{N}\|\boldsymbol{y} - w\boldsymbol{x}\|_2^2$$

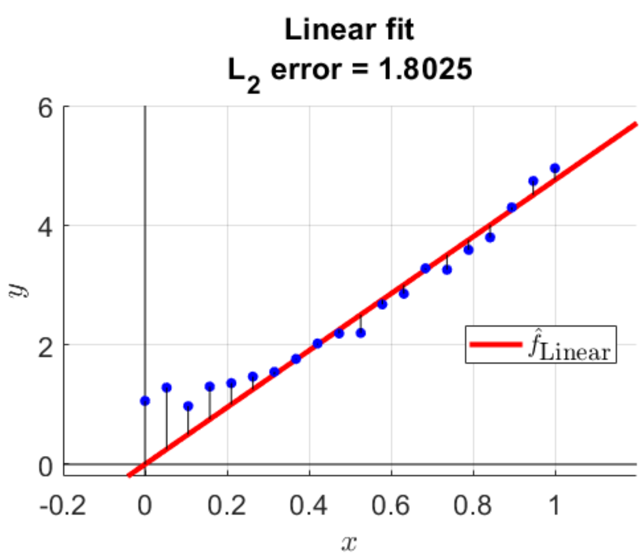
where:

$$\boldsymbol{y} \triangleq \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \boldsymbol{x} \triangleq \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$$

we can find the optimal w be comparing the derivative to zero:

$$\begin{aligned} \frac{d}{dw}L(\omega) &= 0 \\ \frac{d}{dw}\left(\frac{1}{N}\|\boldsymbol{y} - w\boldsymbol{x}\|_2^2\right) &= 0 \\ -\frac{2}{N}\boldsymbol{x}^T(\boldsymbol{y} - w\boldsymbol{x}) &= 0 \\ \boldsymbol{x}^T\boldsymbol{y} - w\|\boldsymbol{x}\|_2^2 &= 0 \end{aligned}$$

$$\Rightarrow \boxed{w = \frac{\boldsymbol{x}^T\boldsymbol{y}}{\|\boldsymbol{x}\|_2^2}}$$



2.1.2 Affine fit

The linear fit $\hat{f}(x) = wx$ constrains $\hat{f}$ to go through the origin $\hat{f}(0) = 0$.
We can use an affine fit to remove this constraint:

$$\hat{f}_{\text{Affine}}(x) = wx + b$$

To obtain the optimal $w \in \mathbb{R}$ and $b \in \mathbb{R}$ (in L_2 error sense) we write:

$$\tilde{\boldsymbol{x}}_i \triangleq \begin{bmatrix} 1 \\ x_i \end{bmatrix}, \qquad \tilde{\boldsymbol{w}} \triangleq \begin{bmatrix} b \\ w \end{bmatrix}$$

$$\Rightarrow \hat{f}_{\text{Affine}}(x) = wx + b = \tilde{\boldsymbol{x}}^T \tilde{\boldsymbol{w}}$$

$$\Rightarrow L(w,b) = L(\tilde{\boldsymbol{w}}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{f}(x_i)\right)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \tilde{\boldsymbol{x}}_i^T \tilde{\boldsymbol{w}}\right)^2 = \frac{1}{N}\left\|\boldsymbol{y} - \boldsymbol{X}\tilde{\boldsymbol{w}}\right\|_2^2$$

where:

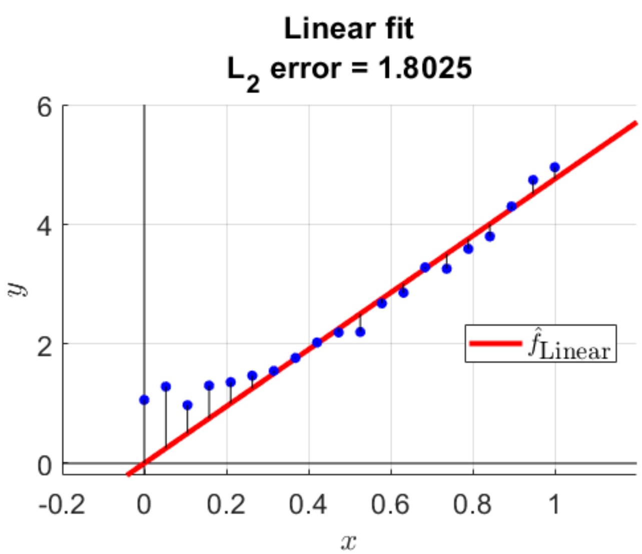
$$\boldsymbol{X} \triangleq \begin{bmatrix} | & & | \\ \tilde{\boldsymbol{x}}_1 & \cdots & \tilde{\boldsymbol{x}}_N \\ | & & | \end{bmatrix}^T = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix}$$

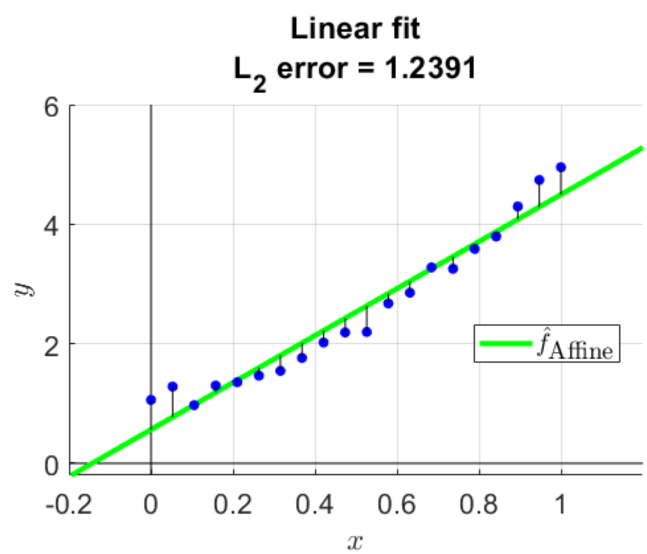
Comparing the gradient to zero:

$$\begin{aligned} \nabla_{\tilde{\boldsymbol{w}}} L(\tilde{\boldsymbol{w}}) &= \boldsymbol{0} \\ \nabla_{\tilde{\boldsymbol{w}}} \left\|\boldsymbol{y} - \boldsymbol{X}\tilde{\boldsymbol{w}}\right\|_2^2 &= \boldsymbol{0} \\ \boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\tilde{\boldsymbol{w}}) &= \boldsymbol{0} \\ \boldsymbol{X}^T\boldsymbol{X}\tilde{\boldsymbol{w}} &= \boldsymbol{X}^T\boldsymbol{y} \end{aligned}$$

$$\Rightarrow \boxed{\tilde{\boldsymbol{w}} = \underbrace{\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T}_{\triangleq \boldsymbol{X}^\dagger}\boldsymbol{y} = \boldsymbol{X}^\dagger \boldsymbol{y}}$$

where $\boldsymbol{X}^\dagger$ is the Moore–Penrose inverse of $\boldsymbol{X}$.





2.1.3 Polyfit

We can assume an M order polynomial model:

$$\hat{f}(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{m=0}^{M} w_m x^m$$

To obtain the optimal $\{w_m \in \mathbb{R}\}_{m=1}^{M}$ (in L_2 error sense) we write:

$$\boldsymbol{w} \triangleq \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix}, \qquad \boldsymbol{\phi}(x) \triangleq \begin{bmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^M \end{bmatrix}$$

$$\Rightarrow \hat{f}(x) = \boldsymbol{\phi}^T(x) \cdot \boldsymbol{w}$$

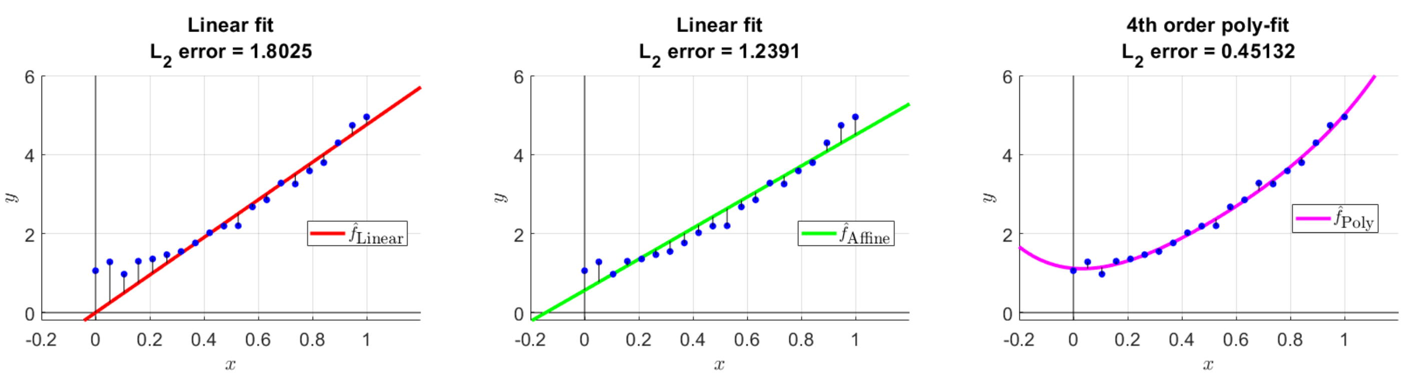
$$\Rightarrow L(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{f}(x_i)\right)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \boldsymbol{\phi}^T(x_i)\,\boldsymbol{w}\right)^2 = \frac{1}{N}\left\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w}\right\|_2^2$$

where:

$$\boldsymbol{\Phi} \triangleq \begin{bmatrix} | & & | \\ \boldsymbol{\phi}(x_1) & \cdots & \boldsymbol{\phi}(x_N) \\ | & & | \end{bmatrix}^T = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^M \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_N & x_N^2 & \dots & x_N^M \end{bmatrix}$$

The loss function is the same as before. Thus, the optimal $\boldsymbol{w}$ is given by:

$$\Rightarrow \boxed{\boldsymbol{w} = \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{y} = \boldsymbol{\Phi}^{\dagger}\boldsymbol{y}}$$



2.1.4 Phase estimation example using feature transform

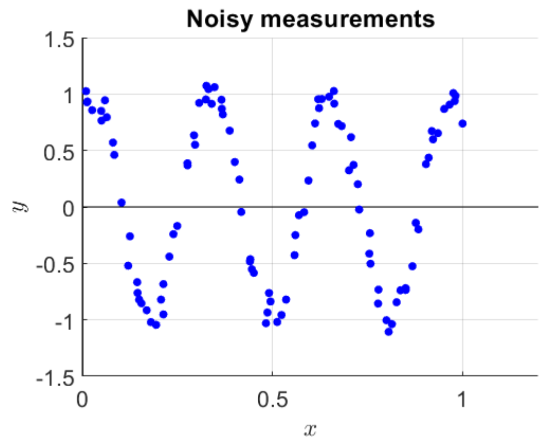
Consider the following function:

$$f(x) = \sin(\omega_0 x + \theta)$$

where ω_0 is known and θ is the parameter we want to estimate.
We obtain N noisy measurements:

$$y_i = f(x_i) + n_i, \qquad i = 1, 2, \ldots, N$$

where n_i is some random noise (assume zero mean).



Estimate θ.
Solution:
Note that (trigonometric identity):

$$\sin(\omega_0 x + \theta) = \sin(\omega_0 x)\cos(\theta) + \cos(\omega_0 x)\sin(\theta)$$

and consider the following feature transform:

$$\boxed{\boldsymbol{\phi}(x) \triangleq \begin{bmatrix} \sin(\omega_0 x) \\ \cos(\omega_0 x) \end{bmatrix}}$$

Let us denote:

$$\boxed{\boldsymbol{w} \triangleq \begin{bmatrix} \cos\left(\hat{\theta}\right) \\ \sin\left(\hat{\theta}\right) \end{bmatrix}}$$

$$\Rightarrow \hat{f}(x) = \sin\left(\omega_0 x + \hat{\theta}\right) = \sin(\omega_0 x)\cos(\theta) + \cos(\omega_0 x)\sin(\theta) = \boldsymbol{\phi}^T(x)\,\boldsymbol{w}$$

As before, the loss function is given by:

$$L(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{f}(x_i)\right)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \boldsymbol{\phi}^T(x_i)\,\boldsymbol{w}\right)^2 = \frac{1}{N}\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w}\|_2^2$$

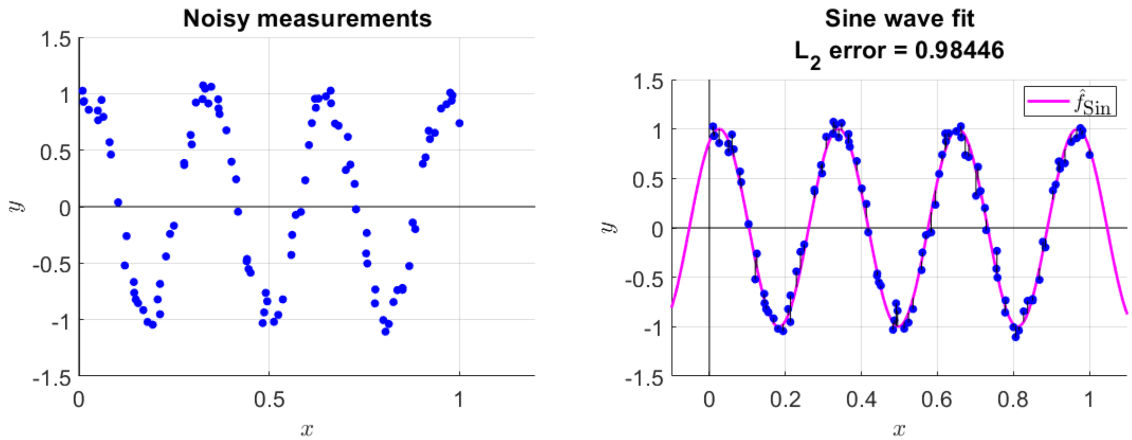
where:

$$\boldsymbol{\Phi} \triangleq \begin{bmatrix} | & & | \\ \boldsymbol{\phi}(x_1) & \cdots & \boldsymbol{\phi}(x_N) \\ | & & | \end{bmatrix}^T = \begin{bmatrix} \sin(\omega_0 x_1) & \cos(\omega_0 x_1) \\ \vdots & \vdots \\ \sin(\omega_0 x_N) & \cos(\omega_0 x_N) \end{bmatrix}$$

Since this is exactly the same loss function, the solution is given by:

$$\begin{bmatrix} \cos\left(\hat{\theta}\right) \\ \sin\left(\hat{\theta}\right) \end{bmatrix} = \boldsymbol{w} = \boldsymbol{\Phi}^{\dagger}\boldsymbol{y}$$

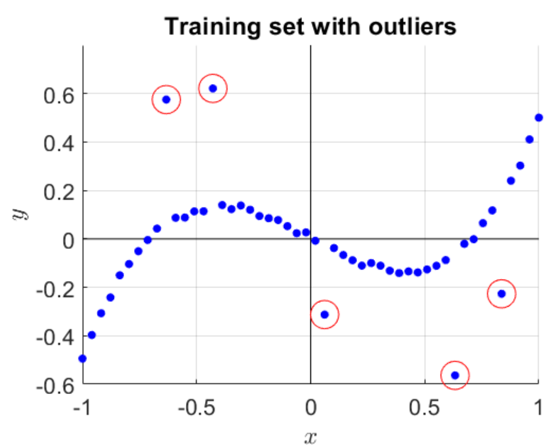
$$\Rightarrow \hat{\theta} = \begin{cases} \arctan\left(\frac{\sin(\hat{\theta})}{\cos(\hat{\theta})}\right) & \cos\left(\hat{\theta}\right) \geq 0 \\ \arctan\left(\frac{\sin(\hat{\theta})}{\cos(\hat{\theta})}\right) + \pi & \cos\left(\hat{\theta}\right) < 0 \end{cases}$$



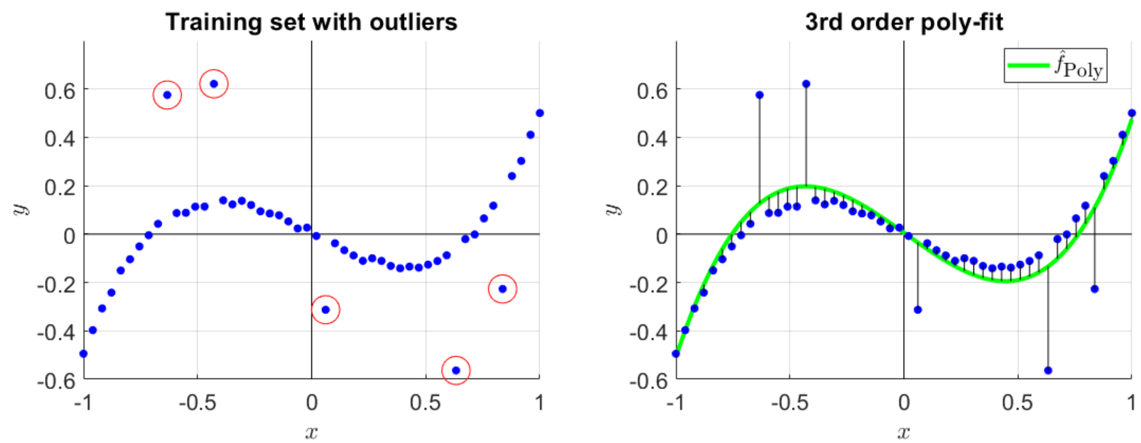
2.2 Outliers and regularization

2.2.1 Outliers

Consider the following training set with few outliers:



If we ignore the outliers the data seems to originated by a 3rd order polynomial.
Thus, we can try to apply poly-fit (of order 3):



The fitted curve is minimizing the squared error:

$$L(\boldsymbol{w}) = \sum_{i=1}^{N} \left(y_i - \hat{f}(x_i)\right)^2 = \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w}\|_2^2$$

Thus, the squared errors of the outliers causing the fitted curve to be too twisted (i.e. the values of $\boldsymbol{w}$ are too big)

2.2.2 Tichonov regularization (Ridge regression)

Regularization can reduce the influence of the outliers.
The loss function with a squared norm regularization (Tichonov regularization) is given by:

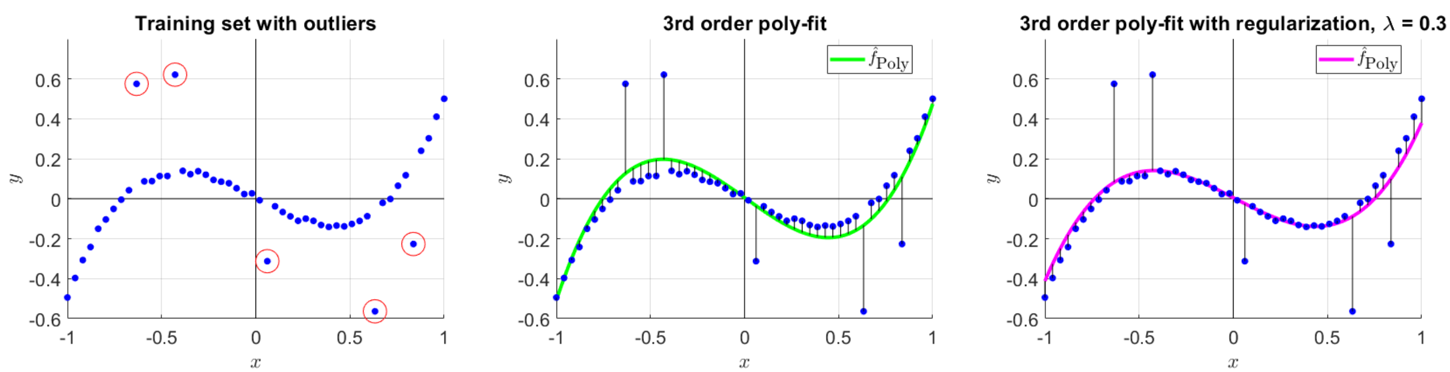
$$\mathcal{L}(\boldsymbol{w}) \triangleq L(\boldsymbol{w}) + \lambda \sum_{m=1}^{M} w_m^2 = \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w}\|_2^2 + \lambda \|\boldsymbol{w}\|_2^2$$

where the parameter λ controls the ratio between the original loss (fidelity) and the regularization term.

To obtain the optimal $\boldsymbol{w}$, we compare the gradient to zero:

$$\begin{aligned}
\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}) &= \boldsymbol{0} \\
-2\boldsymbol{\Phi}^T(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w}) + 2\lambda\boldsymbol{w} &= \boldsymbol{0} \\
\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w} + \lambda\boldsymbol{w} &= \boldsymbol{\Phi}^T\boldsymbol{y} \\
\left(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\boldsymbol{I}\right)\boldsymbol{w} &= \boldsymbol{\Phi}^T\boldsymbol{y}
\end{aligned}$$

$$\Rightarrow \boxed{\boldsymbol{w} = \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{y}}$$



2.2.3 General quadratic regularization

We can put different weights for each parameter of the model $\boldsymbol{w}$:

$$\mathcal{L}(\boldsymbol{w}) = L(\boldsymbol{w}) + \sum_{m=1}^{M} \lambda_m w_m^2 = \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w}\|_2^2 + \|\boldsymbol{\Lambda}\boldsymbol{w}\|_2^2$$

where:

$$\boldsymbol{\Lambda} \triangleq \begin{bmatrix} \lambda_1^{1/2} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_M^{1/2} \end{bmatrix}$$

In this case the optimal parameters are given by:

$$\Rightarrow \boxed{\boldsymbol{w} = \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \boldsymbol{\Lambda}^T\boldsymbol{\Lambda}\right)^{-1} \boldsymbol{\Phi}^T\boldsymbol{y}}$$

Note: $\boldsymbol{\Lambda}$ does not have to be a diagonal matrix.

2.2.4 ℓ_1 - regularization (LASSO)

Another common choice is using the ℓ_1 regularization:

$$\mathcal{L}(\boldsymbol{w}) = \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w}\|_2^2 + \|\boldsymbol{w}\|_1$$

In this case there is no closed form solution but numerical iterative algorithm can provide the optimal value of $\boldsymbol{w}$.
This problem is known as LASSO - Least Absolute Shrinkage and Selection Operator.