Linear Regression

CoE197M/EE298M (Foundations of Machine Learning)

Rowel Atienza, Ph.D.

rowel@eee.upd.edu.ph

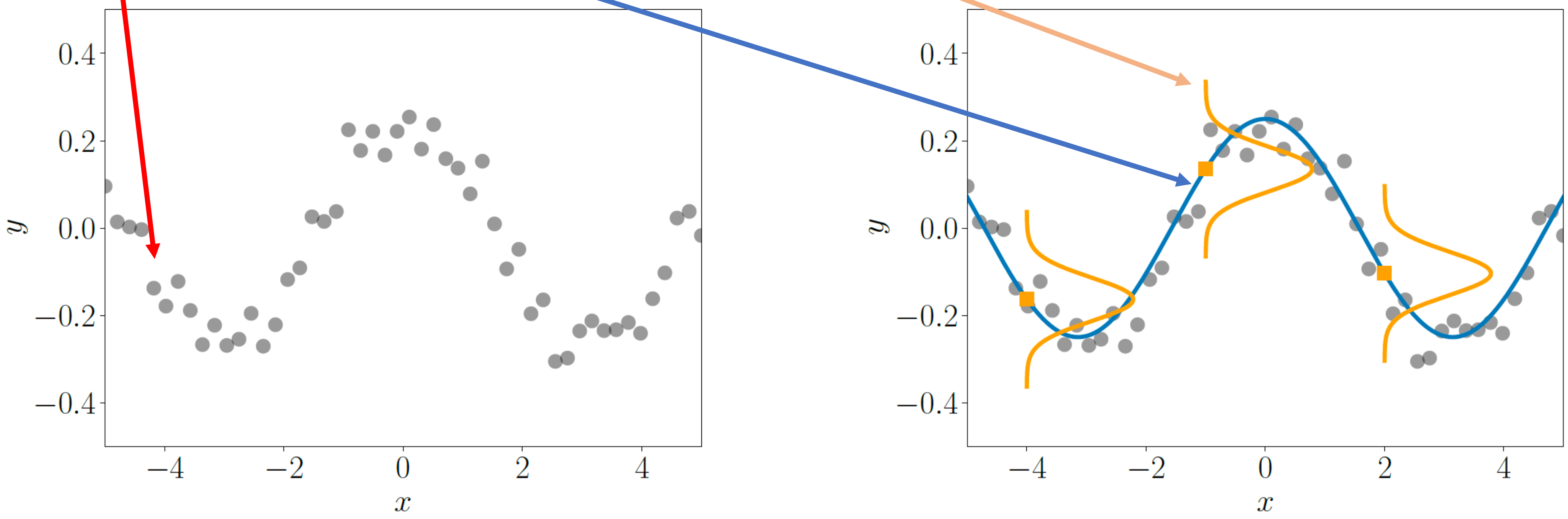
Reference: "Mathematics for Machine Learning". Copyright 2020 by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. Published by Cambridge University Press.

Linear Regression (Curve Fitting)

Input: $\boldsymbol{x}_n$

Output: $y_n = f(\boldsymbol{x}_n) + \epsilon$

Problem: Find $f(\cdot)$



Probabilistic Estimation

$$p(y|\boldsymbol{x}) = \mathcal{N}(y|f(\boldsymbol{x}), \epsilon)$$

$\boldsymbol{x} \in \mathbb{R}^D$, $y \in \mathbb{R}$, and $\epsilon \sim \mathcal{N}(0, \sigma^2)$

Can be rewritten as:

$$y = f(\boldsymbol{x}) + \epsilon$$

Parametric Model

$$p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(y|\boldsymbol{x}^T\boldsymbol{\theta}, \sigma^2)$$

$$y = \boldsymbol{x}^T\boldsymbol{\theta} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

The problem of linear regression boils down to finding the optimal:

$$\boldsymbol{\theta} \in \mathbb{R}^D$$

Note: The function $y = \boldsymbol{x}^T\boldsymbol{\theta}$ is a straight line

Machine Learning is about learning $\boldsymbol{\theta}$ from data

Linear Regression

Linear : It means the parameters $\boldsymbol{\theta}$ are linear but $\boldsymbol{x}$ could be non-linear

Regression : Curve fitting using parameters $\boldsymbol{\theta}$

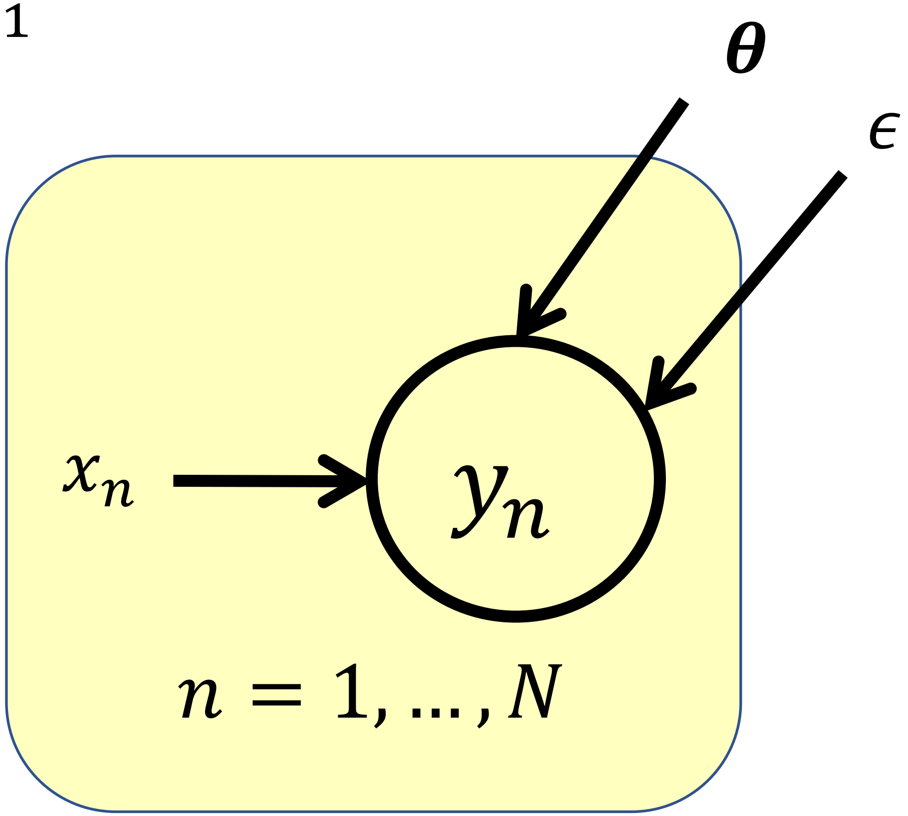
Parameter Estimation

Consider a dataset $\mathcal{D} := \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\} = \{\mathcal{X}, \mathcal{Y}\}$

Each $\boldsymbol{x}_n \in \mathbb{R}^D$ corresponds to $y_n \in \mathbb{R}$

$$p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta}) = p(y_1, \dots, y_N | \boldsymbol{x}_1, \dots, \boldsymbol{x}_N, \boldsymbol{\theta})$$

$$= \prod_{n=1}^{N} p(y_n | \boldsymbol{x}_n^T \boldsymbol{\theta}, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(y_n | \boldsymbol{x}_n^T \boldsymbol{\theta}, \sigma^2)$$



Optimal Point Estimate

Goal is to find the optimal point estimate: $\boldsymbol{\theta}^* \in \mathbb{R}^D$

$$p(y_*|\boldsymbol{x}_*, \boldsymbol{\theta}^*) = \mathcal{N}(y_*|\boldsymbol{x}_*^T\boldsymbol{\theta}^*, \sigma^2)$$

Maximum Likelihood Estimation

MLE

$$\boldsymbol{\theta}_{ML} = \underset{\theta}{\operatorname{argmax}}\, \overbrace{p(\underbrace{\mathcal{Y}|\mathcal{X}}_{\text{Training data}}, \underbrace{\boldsymbol{\theta}}_{\text{Parameters}})}^{\text{Prediction}}$$

Finding Optimal Parameters by Negative Log Likelihood (NLL) Minimization

$$-\log p(\mathcal{Y}|\mathcal{X},\boldsymbol{\theta}) = -\log \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n^T\boldsymbol{\theta}) = -\sum_{n=1}^{N} \log p(y_n|\boldsymbol{x}_n^T\boldsymbol{\theta})$$

If the likelihood is a Gaussian:

$$-\log p(y_n|\boldsymbol{x}_n^T\boldsymbol{\theta}) = \frac{1}{2\sigma^2}(\underbrace{y_n - \boldsymbol{x}_n^T\boldsymbol{\theta}}_{\text{Prediction Error}})^2 + k$$

Loss Function

$$\mathcal{L}(\boldsymbol{\theta}) := \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_n^T \boldsymbol{\theta})^2$$

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2\sigma^2} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^T (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})$$

Where $\boldsymbol{X} := [\boldsymbol{x}_1 \quad \cdots \quad \boldsymbol{x}_N]^T \in \mathbb{R}^{N \times D}$ and $\boldsymbol{y} = [y_1 \quad \cdots \quad y_N]^T \in \mathbb{R}^N$

Minimum

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial}{\partial \boldsymbol{\theta}} \frac{1}{2\sigma^2} (y - \boldsymbol{X\theta})^T (y - \boldsymbol{X\theta})$$

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\frac{1}{\sigma^2} (y - \boldsymbol{X\theta})^T \boldsymbol{X} = -\frac{1}{\sigma^2} (y^T \boldsymbol{X} - \boldsymbol{\theta}^T \boldsymbol{X}^T \boldsymbol{X}) = \boldsymbol{0}^T$$

$$\boldsymbol{\theta}_{ML}^T = \boldsymbol{\theta}^T = y^T \boldsymbol{X} (\boldsymbol{X}^T \boldsymbol{X})^{-1}$$

$$\boldsymbol{\theta}_{ML} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T y$$

$\boldsymbol{X}^T \boldsymbol{X} \in \mathbb{R}^{N \times N}$ is symmetric and must be invertible or $rank(\boldsymbol{X}) = N$

This is the global minimum since the Hessian $\frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2} = \boldsymbol{X}^T \boldsymbol{X}$ is positive definite

Properties used

Theorem (SPSD): For a given matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, we can always obtain a symmetric positive semi-definite matrix $\boldsymbol{S} \in \mathbb{R}^{n \times n}$: $\boldsymbol{S} = \boldsymbol{A}^T \boldsymbol{A}$

If $rank(\boldsymbol{A}) = n$, then $\boldsymbol{S}$ is a symmetric positive definite (SPD) matrix

The inverse of a symmetric matrix is also symmetric

$$\boldsymbol{S}^{-1} = (\boldsymbol{S}^{-1})^T$$

MLE with Features $\boldsymbol{\phi}(\boldsymbol{x})$

$\phi(\boldsymbol{x})$ is a non-linear transformation of $\boldsymbol{x}$

$$p(y|\boldsymbol{x},\boldsymbol{\theta}) = \mathcal{N}(y|\phi^T(\boldsymbol{x})\boldsymbol{\theta},\sigma^2)$$

$$y = \phi^T(\boldsymbol{x})\boldsymbol{\theta} + \epsilon, \quad \epsilon\sim\mathcal{N}(0,\sigma^2)$$

Example of Non-linear $\phi(x)$

$$\phi(x) = \begin{bmatrix} \phi_0(x) \\ \phi_1(x) \\ \vdots \\ \phi_{K-1}(x) \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ \vdots \\ x^{K-1} \end{bmatrix} \in \mathbb{R}^K$$

A polynomial of degree $K - 1$ can be expressed:

$$f(x) = \sum_{k=1}^{K-1} \theta_k x^k = \phi^T(x)\boldsymbol{\theta}$$

Feature Matrix

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi^T(\boldsymbol{x}_1) \\ \vdots \\ \phi^T(\boldsymbol{x}_N) \end{bmatrix} = \begin{bmatrix} \phi_0(\boldsymbol{x}_1) & \cdots & \phi_{K-1}(\boldsymbol{x}_1) \\ & \vdots & \\ \phi_0(\boldsymbol{x}_N) & \cdots & \phi_{K-1}(\boldsymbol{x}_N) \end{bmatrix}$$

$$\boldsymbol{\Phi}_{ij} = \phi_j(\boldsymbol{x}_i), \phi_j : \mathbb{R}^D \to \mathbb{R}$$

Feature Matrix of 2nd Order Polynomial

$$\mathbf{\Phi} = \begin{bmatrix} \phi^T(\boldsymbol{x}_1) \\ \vdots \\ \phi^T(\boldsymbol{x}_N) \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 \\ & \vdots & \\ 1 & x_N & x_N^2 \end{bmatrix}$$

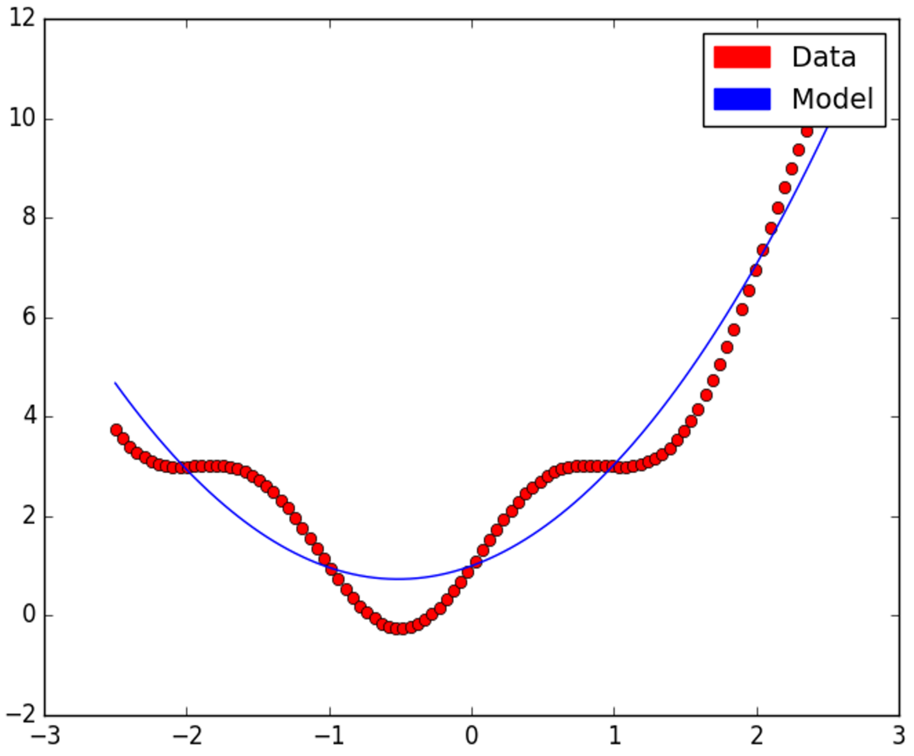
NLL with Feature Matrix

$$-\log p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta}) = \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{\Phi\theta})^T(\boldsymbol{y} - \boldsymbol{\Phi\theta})$$

MLE for Feature Matrix:

$$\boldsymbol{\theta}_{ML} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{y}$$

$\boldsymbol{\Phi}^T\boldsymbol{\Phi} \in \mathbb{R}^{K\times K}$ must be invertible or $rank(\boldsymbol{\Phi}) = K$



Distribution Function:

Output is second degree polynomial:

$$y = x^2 + x + 1$$

Sinusoidal noise is added to output.

Estimating Noise Variance

MLE for Estimating Noise Variance $\boldsymbol{\sigma_{ML}^2}$

$$\log p(\mathcal{Y}|\mathcal{X},\boldsymbol{\theta},\sigma^2) = \sum_{n=1}^{N} \log \mathcal{N}(y_n|\phi^T(\boldsymbol{x}_n)\boldsymbol{\theta},\sigma^2)$$

$$= \sum_{n=1}^{N} \left(-\frac{1}{2}\log 2\pi - \frac{1}{2}\log \sigma^2 - \frac{1}{2\sigma^2}(y_n - \phi^T(x_n)\boldsymbol{\theta})^2\right)$$

$$= -\frac{N}{2}\log \sigma^2 - \frac{1}{2\sigma^2}\underbrace{\sum_{n=1}^{N}(y_n - \phi^T(x_n)\boldsymbol{\theta})^2}_{s} + k$$

MLE of $\boldsymbol{\sigma}^2$ is the mean of squared distances between empirical observation and prediction

$$\frac{d \log p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta}, \sigma^2)}{d\sigma^2} = -\frac{N}{2\sigma^2} + \frac{s}{2\sigma^4} = 0$$

$$\sigma_{ML}^2 = \frac{s}{N} = \frac{1}{N}\sum_{n=1}^{N}(y_n - \phi^T(x_n)\boldsymbol{\theta})^2$$

Capacity and Overfitting

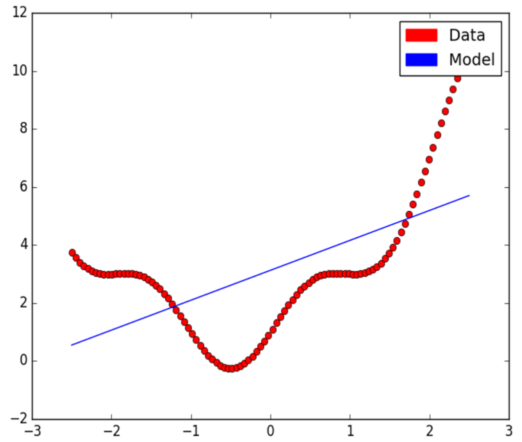
Capacity

Capacity - ability to fit a wide variety of functions

↓ Capacity → Underfitting: ↑ Train Error , ↑ Test Error

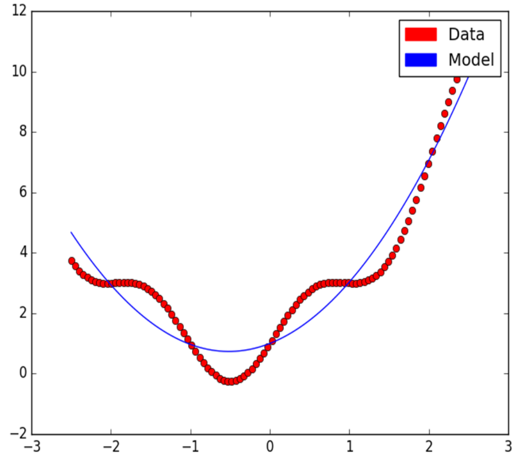
↑ Capacity → Overfitting: ↓ Train Error , ↑ Test Error

✓ Capacity → Optimal Fit: ↓ Train Error , ↓ Test Error



Underfitting: 1st degree polynomial

Optimal Fit: 2nd degree



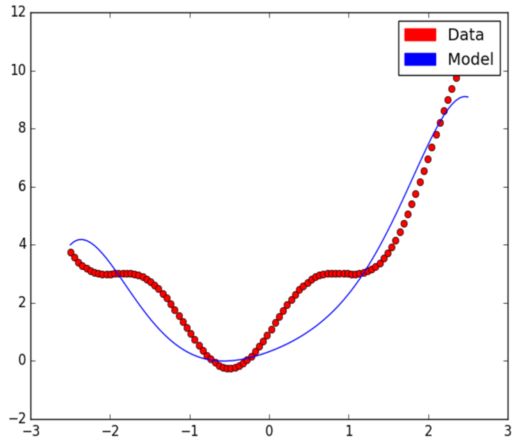
Distribution Function:

Output is second degree polynomial:

$y = x^2 + x + 1$

Sinusoidal noise is added to output.

Overfitting: 6th degree



MLE is susceptible to overfitting

Given over capacity, MLE can easily memory the dataset resulting to overfitting

$$-\log p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta}) = \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{\Phi\theta})^T(\boldsymbol{y} - \boldsymbol{\Phi\theta})$$

Maximum A Posteriori (MAP) Estimation

Motivation

MLE is susceptible to overfitting

MAP maximizes the posterior given a dataset:

$$p(\boldsymbol{\theta}|\mathcal{X},\mathcal{Y}) = \frac{p(\mathcal{Y}|\mathcal{X},\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{Y}|\mathcal{X})}$$

The prior $p(\boldsymbol{\theta})$ has influence on the posterior $p(\boldsymbol{\theta}|\mathcal{X},\mathcal{Y})$

The parameter vector that maximizes the posterior is called MAP estimate

MAP

$$\log p(\boldsymbol{\theta}|\mathcal{X}, \mathcal{Y}) = \log p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) + k$$

NLL in MAP

$$\boldsymbol{\theta}_{MAP} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}(-\log p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta}) - \log p(\boldsymbol{\theta}))$$

NLL in MAP

$$-\frac{d\log p(\boldsymbol{\theta}|\mathcal{X},\mathcal{Y})}{d\boldsymbol{\theta}} = \frac{d(\overbrace{-\log p(\mathcal{Y}|\mathcal{X},\boldsymbol{\theta})}^{\text{MLE}} - \log p(\boldsymbol{\theta}))}{d\boldsymbol{\theta}}$$

Recall: $-\log p(\mathcal{Y}|\mathcal{X},\boldsymbol{\theta}) = \frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{\Phi\theta})^T(\boldsymbol{y}-\boldsymbol{\Phi\theta})$

Assume: $p(\boldsymbol{\theta}) = \mathcal{N}(0, b^2)$

Then: $-\log p(\boldsymbol{\theta}) = \frac{1}{2b^2}\boldsymbol{\theta}^T\boldsymbol{\theta}$

$\boldsymbol{\theta}_{MAP}$ is found by setting $\dfrac{d \log p(\boldsymbol{\theta}|\mathcal{X},\mathcal{Y})}{d\boldsymbol{\theta}} = \mathbf{0}^T$

$$-\frac{d \log p(\boldsymbol{\theta}|\mathcal{X},\mathcal{Y})}{d\boldsymbol{\theta}} = \frac{1}{\sigma^2}(\boldsymbol{\theta}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi} - y^T\boldsymbol{\Phi}) + \frac{1}{b^2}\boldsymbol{\theta}^T = \mathbf{0}^T$$

$$\boldsymbol{\theta}^T\left(\frac{1}{\sigma^2}\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \frac{1}{b^2}\boldsymbol{I}\right) - \frac{1}{\sigma^2}\boldsymbol{y}^T\boldsymbol{\Phi} = \mathbf{0}^T$$

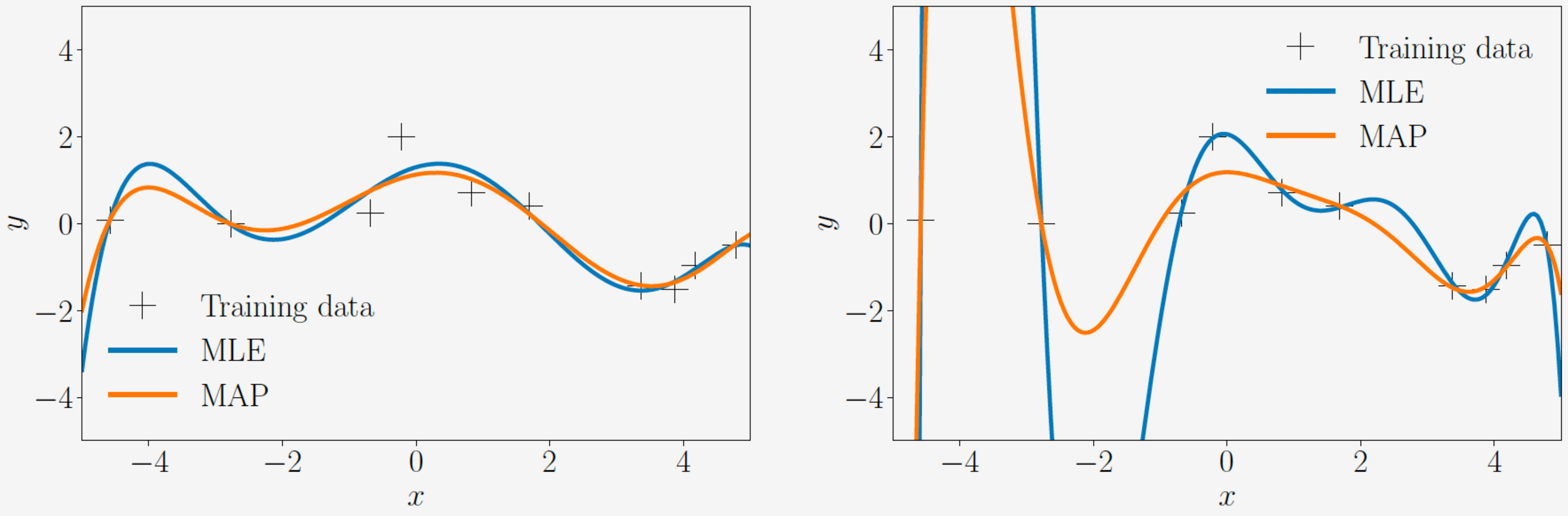
$$\boldsymbol{\theta}^T\left(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \frac{\sigma^2}{b^2}\boldsymbol{I}\right) = \boldsymbol{y}^T\boldsymbol{\Phi}$$

$$\boldsymbol{\theta}^T = \boldsymbol{y}^T\boldsymbol{\Phi}\left(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \frac{\sigma^2}{b^2}\boldsymbol{I}\right)^{-1}$$

$$\boldsymbol{\theta}_{MAP} = \left(\underbrace{\boldsymbol{\Phi}^T\boldsymbol{\Phi}} + \underbrace{\frac{\sigma^2}{b^2}\boldsymbol{I}}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{y}$$

Symmetric Positive Semi-Definite + Symmetric Positive Definite =Positive Definite

MLE vs MAP



(a) Polynomials of degree 6.   (b) Polynomials of degree 8.

MAP as a Regularizer

Instead of assuming $p(\boldsymbol{\theta}) = \mathcal{N}(0, b^2)$, we can assume a generalized regularization term added to the MLE:

$$\mathcal{L}(\boldsymbol{\theta}) = \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta}\|^2 + \lambda\|\boldsymbol{\theta}\|_p^2$$

Where $p = 1,2, \dots, P$

Note: When $p = 2$, $\mathcal{L}(\boldsymbol{\theta})$ is a MAP loss function (L2 regularization)

Note: When $p = 1$, $\mathcal{L}(\boldsymbol{\theta})$ is L1 regularized MLE

To be continued…