Principles of Machine Learning

Lecture 4: Linear Regression

Sharif University of Technology
Dept. of Aerospace Engineering

March 11, 2025

Table of Contents

Table of Contents

Linear Regression

- A very widely-used method for predicting a real-valued output (**dependent variable** or **target**) $y \in \mathbb{R}$, given a vector of real-valued input (**independent variables** or **explanatory variables**, or **covariates**) $x \in \mathbb{R}^{\mathrm{D}}$
- Key property: We assume that $\mathbb{E}[y|\mathbf{x}] = \boldsymbol{\theta}^{\top}\mathbf{x}$

Linear Regression

- **Choice of the model (type) and the parametrization**
  Function classes (e.g., polynomials) and particular parametrization (e.g., degree of the polynomial)
- **Finding good parameters**
  Loss/objective function and an optimization algorithm to minimize it
- **Overfitting and model selection**
  Typically occurs if the underlying model (or its parametrization) is overly flexible and expressive
- **Relationship between loss functions and parameter priors**
  Connection between loss functions and the underlying prior assumptions that induce these losses
- **Uncertainty modeling**
  Describe the remaining parameter uncertainty to obtain a measure of confidence of the model's prediction at test time

Linear Regression Applications

- Time-series analysis (e.g., system identification)
- Control and robotics (e.g., reinforcement learning, forward/inverse model learning)
- Optimization (e.g., line searches, global optimization)
- Deep learning applications (e.g., computer games, speech-to-text translation, image recognition, automatic video annotation)

Linear Regression

We aim to find a function f that maps inputs $\mathbf{x} \in \mathbb{R}^D$ to corresponding function values $f(\mathbf{x}) \in \mathbb{R}$.

We are given training inputs $\mathbf{x}_n$ and corresponding noisy observations $y_n = f(\mathbf{x}_n) + \epsilon$, where ϵ is an i.i.d random variable that describes measurement/observation noise.

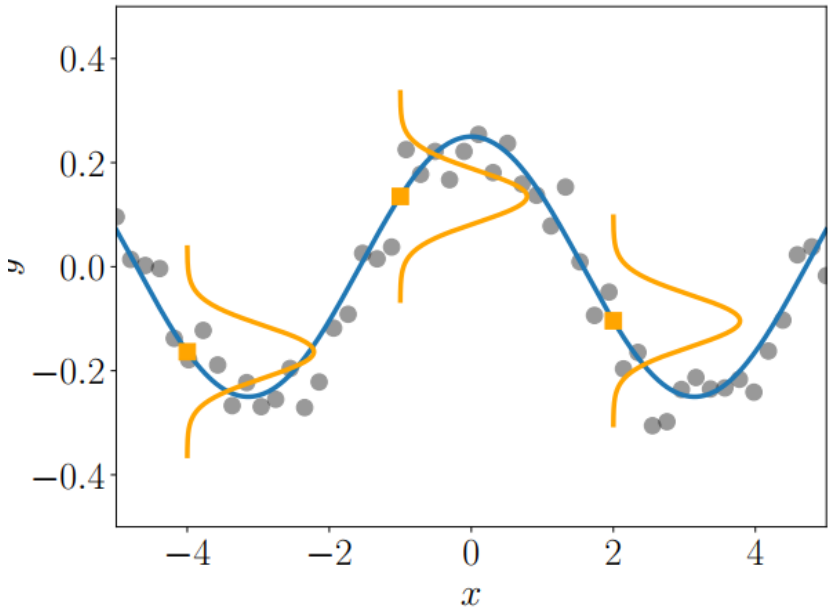


Table of Contents

Standard Linear Regression – Problem Formulation

Presence of observation noise $\rightarrow$ Probabilistic approach $\rightarrow$ Explicitly model the noise using a likelihood function:

$$p(y \mid \mathbf{x}) = \mathcal{N}(y \mid f(\mathbf{x}), \sigma^2).$$

with the input-output relationship given as:

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

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is i.i.d. Gaussian measurement noise with mean 0 and variance σ^2.

Our objective is to find a function that is similar to the unknown function f that generated the data and that generalizes well.

Standard Linear Regression – Problem Formulation

For now, assume that noise variance σ^2 is known and focus on learning the model parameters $\boldsymbol{\theta}$, where in linear regression, $\boldsymbol{\theta}$ appear linearly in our model:

$$p(y \mid \mathbf{x}, \theta) = \mathcal{N}(y \mid \mathbf{x}^\top \boldsymbol{\theta}, \sigma^2) \tag{1}$$

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

where $\theta \in \mathbb{R}^D$ are the parameters we seek.

- The *likelihood* in (1) is the probability density function of y evaluated at $\mathbf{x}^\top \boldsymbol{\theta}$.
- Without the source of uncertainty (noise), (1) would be a Dirac delta.

Standard Linear Regression – Problem Formulation

- The linear regression model (2) is linear in the parameters $\boldsymbol{\theta}$ and the inputs $\mathbf{x}$.
- Note that $y = \boldsymbol{\phi}^\top(\mathbf{x})\boldsymbol{\theta}$ is also a linear regression model for nonlinear transformations $\boldsymbol{\phi}$. (Why?)

In the following, we discuss how to find good parameters $\boldsymbol{\theta}$ and how to evaluate whether or not they work well.
For now, we assume that σ^2 is known.

Table of Contents

Parameter Estimation

Given a *training set* $\mathcal{D} := \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ and noting that y_i and y_j are conditionally independent, the likelihood is:

$$p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) = p(y_1, \ldots, y_N \mid \mathbf{x}_1, \ldots, \mathbf{x}_N, \boldsymbol{\theta}) \tag{3}$$

$$= \prod_{n=1}^{N} p(y_n \mid \mathbf{x}_n, \boldsymbol{\theta}) = \prod_{n=1}^{N} \mathcal{N}(y_n \mid \mathbf{x}_n^\top \boldsymbol{\theta}, \sigma^2), \tag{4}$$

where $\mathcal{X} := \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and $\mathcal{Y} := \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$.

- The likelihood and the factors $p(y_n \mid \mathbf{x}_n, \boldsymbol{\theta})$ are Gaussian due to the noise distribution.

Parameter Estimation

Parameter estimation problem: find optimal parameters $\boldsymbol{\theta}^* \in \mathbb{R}^D$ for the linear regression model (3).

Once the $\boldsymbol{\theta}^*$ are found $\rightarrow$ predict function values by using this parameter estimate in (3):

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

where $\mathbf{x}_*$ is an arbitrary test input and y_* is the distribution of the corresponding target at that input.

How to solve this parameter estimation problem / find $\boldsymbol{\theta}^*$?

Maximum Likelihood Estimation (MLE)

Maximum Likelihood Estimation: finding $\boldsymbol{\theta}_{ML}$ that maximize the likelihood (4).

- Maximizing the likelihood $\rightarrow$ maximizing the predictive distribution of the training data given the model params:

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}). \tag{5}$$

- The likelihood $p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$ is not a probability distribution in $\boldsymbol{\theta}$.
- It is simply a function of the parameters $\boldsymbol{\theta}$ but does not integrate to 1 and may not even be integrable with respect to $\boldsymbol{\theta}$.
- However, the likelihood in (5) is a normalized probability distribution in $\mathbf{y}$.

Maximum Likelihood Estimation (MLE)

To find $\boldsymbol{\theta}_{ML}$:

1. Gradient descent (or GD on the negative likelihood).
2. Apply log-transformation to the likelihood function & Minimize the negative log-likelihood
   - Logarithm is a (strictly) monotonically increasing function $\rightarrow$ the optimum of f is identical to optimum of $\log f$

MLE in linear regression has closed-form solution $\rightarrow$ Option 2 is preferable.

Log-transformations are useful because:

- does not suffer from numerical underflow
- more simple differentiation rules

Maximum Likelihood Estimation (MLE)

Thus, to find $\boldsymbol{\theta}_{ML}$, we minimize the negative log-likelihood:

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

In the linear regression model (2), the likelihood is Gaussian, such that:

$$\log\, p(y_n\,|\,\mathbf{x}_n,\boldsymbol{\theta}) = -\frac{1}{2\sigma^2}(y_n - \mathbf{x}_n^\top\boldsymbol{\theta})^2 + \text{const},$$

where the constant includes all terms independent of $\boldsymbol{\theta}$.

Maximum Likelihood Estimation (MLE)

The negative log-likelihood function (also called *error function* or *loss function*) is then as follows:

$$\mathcal{L}(\boldsymbol{\theta}) \coloneqq \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\top \boldsymbol{\theta})^2 \qquad (6)$$

$$= \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) = \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2, \qquad (7)$$

where we define the *design matrix* $\mathbf{X} \coloneqq [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times D}$ and $\mathbf{y} \coloneqq [y_1, \ldots, y_N]^\top \in \mathbb{R}^N$

- Note that the sum of squared errors between observations y_n and the corresponding model prediction $\mathbf{x}_n^\top \boldsymbol{\theta}$ equals the squared distance between $\mathbf{y}$ and $\mathbf{X}\boldsymbol{\theta}$,

Maximum Likelihood Estimation (MLE)

Using the results from the previous Lecture (Calculus Basics), we compute the gradient of $\mathcal{L}$ w.r.t. the params:

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}} = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\left(\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\theta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\theta})\right) \tag{8}$$

$$= \frac{1}{2\sigma^2}\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\left(\mathbf{y}^\top\mathbf{y} - 2\mathbf{y}^\top\mathbf{X}\boldsymbol{\theta} + \boldsymbol{\theta}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{\theta}\right) \tag{9}$$

$$= \frac{1}{\sigma^2}\left(-\mathbf{y}^\top\mathbf{X} + \boldsymbol{\theta}^\top\mathbf{X}^\top\mathbf{X}\right) \in \mathbb{R}^{1\times \mathrm{D}}. \tag{10}$$

Maximum Likelihood Estimation (MLE)

The MLE estimator $\boldsymbol{\theta}_{ML}$ solves $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}} = \mathbf{0}$:

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}} = \mathbf{0} \overset{(10)}{\iff} \boldsymbol{\theta}_{\mathrm{ML}}^\top \mathbf{X}^\top \mathbf{X} = \mathbf{y}^\top \mathbf{X} \tag{11}$$

$$\iff \boldsymbol{\theta}_{\mathrm{ML}}^\top = \mathbf{y}^\top \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1} \tag{12}$$

$$\iff \boxed{\boldsymbol{\theta}_{\mathrm{ML}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}}. \tag{13}$$

$y = X\theta$

- Ignoring the possibility of duplicate data $\mathrm{rk}(\mathbf{X}) = D$ if $N \geq D$, i.e., we do not have more parameters than data points.

- We could right-multiply the first equation by $(\mathbf{X}^\top \mathbf{X})^{-1}$ because $\mathbf{X}^\top \mathbf{X}$ is positive definite if $\mathrm{rk}(\mathbf{X}) = D$.

Maximum Likelihood Estimation (MLE)

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}} = \mathbf{0} \Longrightarrow \boxed{\boldsymbol{\theta}_{\mathrm{ML}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}}$$

- Setting the gradient to $\mathbf{0}$ is a necessary and sufficient condition to obtain a global minimum since the Hessian $\nabla^2_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \mathbf{X}^\top \mathbf{X} \in \mathbb{R}^{D \times D}$ is positive definite.

- The maximum likelihood solution above requires solving a system of linear equations of the form $\mathbf{A}\boldsymbol{\theta} = \mathbf{b}$ with $\mathbf{A} = (\mathbf{X}^\top \mathbf{X})$ and $\mathbf{b} = \mathbf{X}^\top \mathbf{y}$.

Maximum Likelihood Estimation (MLE) with Features

We considered the linear regression model as (4) that allowed us to fit straight lines to data using MLE.
However, the straight lines are not sufficiently expressive!

Fitting nonlinear functions within the linear regression framework by performing an arbitrary nonlinear transformation $\boldsymbol{\phi}(\mathbf{x})$ of the inputs $\mathbf{x}$:

$$p(y \,|\, \mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y \,|\, \boldsymbol{\phi}^\top(\mathbf{x})\boldsymbol{\theta}, \sigma^2) \tag{14}$$

$$\iff y = \boldsymbol{\phi}^\top(\mathbf{x})\boldsymbol{\theta} + \epsilon = \sum_{\mathrm{k}=0}^{\mathrm{K}-1} \theta_{\mathrm{k}} \phi_{\mathrm{k}}(\mathbf{x}) + \epsilon, \tag{15}$$

where $\boldsymbol{\phi} : \mathbb{R}^D \to \mathbb{R}^K$ is a (nonlinear) transformation of the inputs $\mathbf{x}$ and $\phi_{\mathrm{k}} : \mathbb{R}^D \to \mathbb{R}$ is the kth component of the *feature vector* $\boldsymbol{\phi}$.

Maximum Likelihood Estimation (MLE) with Features

Example 1. Polynomial Regression

A regression problem: $y = \boldsymbol{\phi}^\top(x)\boldsymbol{\theta} + \epsilon$, where $x \in \mathbb{R}$ and $\boldsymbol{\theta} \in \mathbb{R}^K$. A transformation often used in this context is:

$$\boldsymbol{\phi}(x) = \begin{bmatrix} \phi_0(x) \\ \phi_1(x) \\ \vdots \\ \phi_{K-1}(x) \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \\ \vdots \\ x^{K-1} \end{bmatrix} \in \mathbb{R}^K.$$

Then, a polynomial of degree $K-1$ is

$$f(x) = \sum_{k=0}^{K-1} \theta_k x^k = \boldsymbol{\phi}^\top(x)\boldsymbol{\theta},$$

where $\boldsymbol{\phi}$ is defined as above and $\boldsymbol{\theta} = [\theta_0, \ldots, \theta_{K-1}]^\top \in \mathbb{R}^K$ contains the (linear) parameters θ

Maximum Likelihood Estimation (MLE) with Features

Revisiting the design matrix and defining it as the *feature matrix* in this context as:

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi^\top(\mathbf{x}_1) \\ \vdots \\ \phi^\top(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \dots & \phi_{K-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \dots & \phi_{K-1}(\mathbf{x}_2) \\ \vdots & & \vdots \\ \phi_0(\mathbf{x}_N) & \dots & \phi_{K-1}(\mathbf{x}_N) \end{bmatrix} \in \mathbb{R}^{N \times K},$$

where $\Phi_{ij} = \phi_j(\mathbf{x}_i)$ and $\phi_j : \mathbb{R}^D \to \mathbb{R}$.

Example 2. Feature Matrix for Second-order Polynomials

For a second-order polynomial and N training points $x_n \in \mathbb{R}, n = 1, \dots, N$, the feature matrix is

$$\boldsymbol{\Phi} = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 \end{bmatrix}.$$

Maximum Likelihood Estimation (MLE) with Features

With the feature matrix $\mathbf{\Phi}$ defined, the negative log-likelihood for the linear regression model (14,15) can be written as:

$$-\log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}) = \frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) + \text{const.}$$

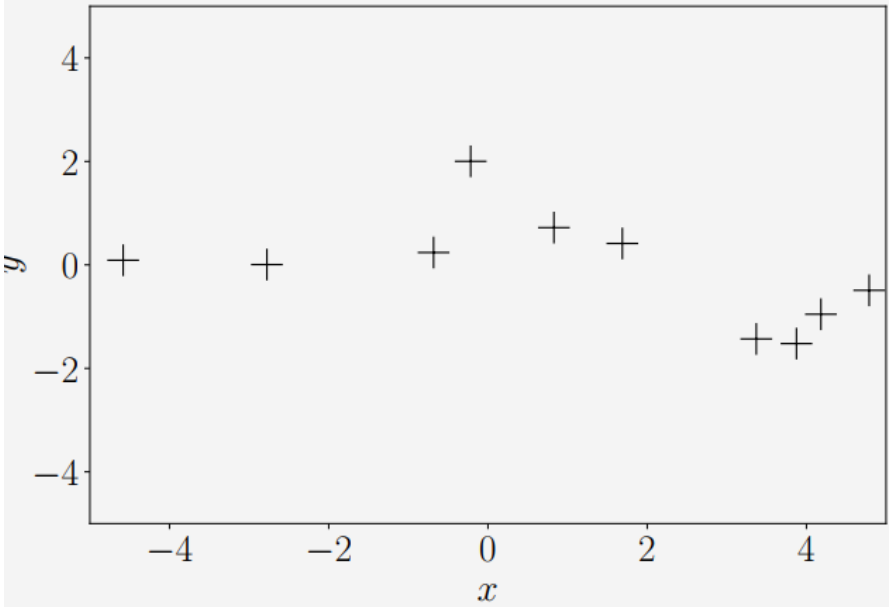
Since both $\mathbf{X}$ and $\mathbf{\Phi}$ are independent of the params $\boldsymbol{\theta}$ that we wish to optimize, the *maximum likelihood estimate* is

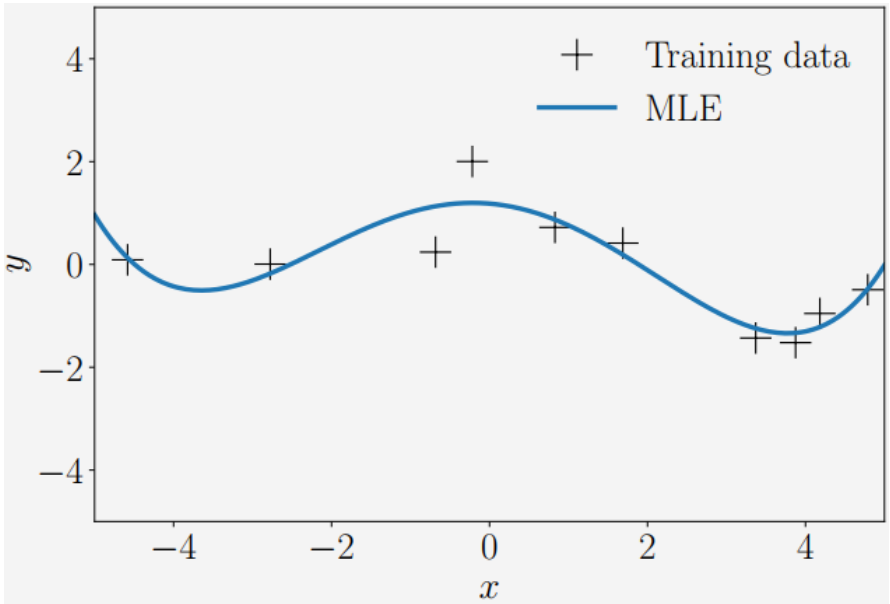
$$\boxed{\boldsymbol{\theta}_{\text{ML}} = (\mathbf{\Phi}^\top \mathbf{\Phi})^{-1}\mathbf{\Phi}\mathbf{y}} \tag{16}$$

for the linear regression problem with nonlinear features defined in (14,15).

Maximum Likelihood Estimation (MLE) with Features

Example 3. Maximum Likelihood Polynomial Fit





The figure (a) shows the dataset consisting of $N = 10$ pairs (x_n, y_n), where $x_n \sim \mathcal{U}[-5, 5]$ and $y_n = -\sin(x_n/5) + \cos(x_n) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 0.2^2)$.

We fit a polynomial of degree 4 using MLE, i.e., parameters $\boldsymbol{\theta}_{ML}$ are given in (16). The result is shown in figure (b).

Maximum Likelihood Estimation (MLE) with Features

To estimate the noise variance σ^2:

$$\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(y-\phi^\top\theta)^2}{2\sigma^2}\right]$$

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

$$= \sum_{n=1}^{N}(-\frac{1}{2}\log(2\pi) - \frac{1}{2}\log\sigma^2 - \frac{1}{2\sigma^2}(y_n - \boldsymbol{\phi}^\top(\mathbf{x}_n)\boldsymbol{\theta})^2)$$

$$= -\frac{N}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\underbrace{\sum_{n=1}^{N}(y_n - \boldsymbol{\phi}^\top(\mathbf{x}_n)\boldsymbol{\theta})^2}_{=:\,s} + \text{const.}$$

Maximum Likelihood Estimation (MLE) with Features

$$\frac{\partial \log\, p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}, \sigma^2)}{\partial \sigma^2} = -\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4} s = 0$$

$$\iff \frac{N}{2\sigma^2} = \frac{s}{2\sigma^4}$$

then

$$\boxed{\sigma^2_{\mathrm{ML}} = \frac{s}{N} = \frac{1}{N} \sum_{n=1}^{N} (y_n - \boldsymbol{\phi}^\top(\mathbf{x}_n)\boldsymbol{\theta})^2}$$

Table of Contents

LSE – MLE as Orthogonal Projection

Recalling linear regression without features:

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

considering a straight line $f : \mathbb{R} \to \mathbb{R}$ that go through origin, the parameter θ determines the slope of the line.
Recalling the maximum likelihood estimator:

$$\theta_{ML} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} = \frac{\mathbf{X}^\top \mathbf{y}}{\mathbf{X}^\top \mathbf{X}} \in \mathbb{R}$$

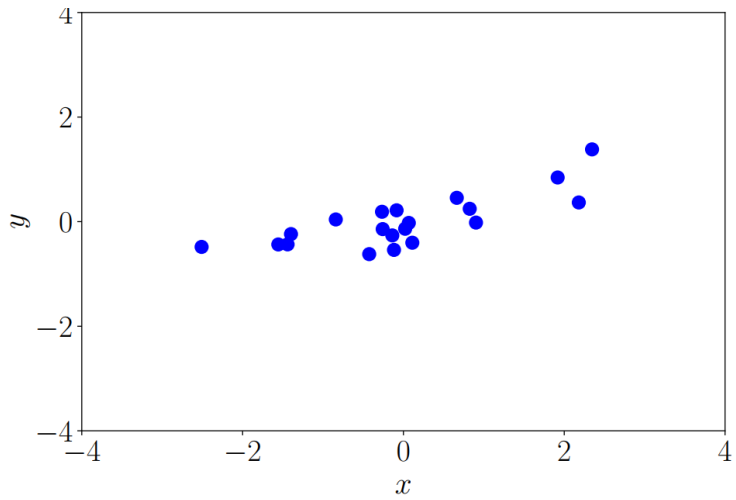
$$\Longrightarrow \mathbf{X}\theta_{ML} = \mathbf{X} \frac{\mathbf{X}^\top \mathbf{y}}{\mathbf{X}^\top \mathbf{X}} = \frac{\mathbf{X}\mathbf{X}^\top}{\mathbf{X}^\top \mathbf{X}} \mathbf{y}$$

We obtain the approximation with the minimum least-squares error between $\mathbf{y}$ and $\mathbf{X}\theta$.

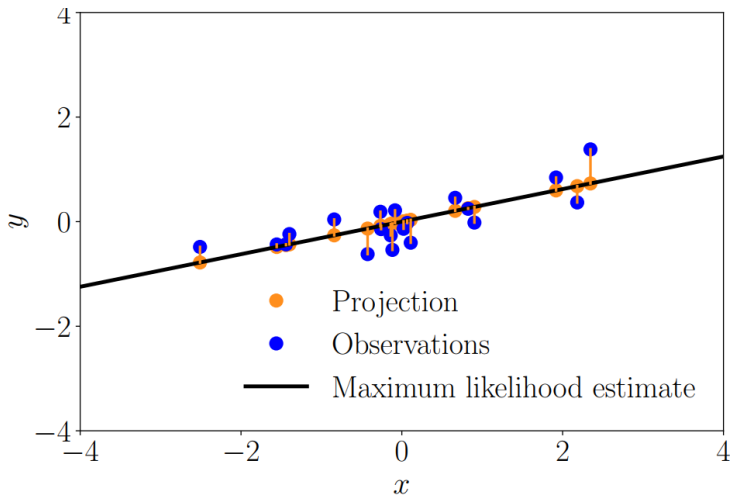
LSE – MLE as Orthogonal Projection

- Linear regression can be thought of as a method for solving systems of linear equations $\mathbf{y} = \mathbf{X}\theta$.
- The ML estimator θ_{ML} effectively does an orthogonal projection of $\mathbf{y}$ onto the 1-D subspace spanned by $\mathbf{X}$.
- $\frac{\mathbf{X}\mathbf{X}^\top}{\mathbf{X}^\top\mathbf{X}}$ is the projection matrix.
- θ_{ML} is the coordinates of the projection onto the 1-D subspace of $\mathbb{R}^N$ spanned by $\mathbf{X}$ and $\mathbf{X}\theta_{\mathrm{ML}}$ as the orthogonal projection of $\mathbf{y}$ onto this subspace.

Geometric interpretation of least squares



(a) Regression dataset consisting of noisy observations y_n (blue) of function values $f(x_n)$ at input locations x_n.



(b) The orange dots are the projections of the noisy observations (blue dots) onto the line $\theta_{\text{ML}}x$. The maximum likelihood solution to a linear regression problem finds a subspace (line) onto which the overall projection error (orange lines) of the observations is minimized.

LSE – MLE as Orthogonal Projection

In the general linear regression case

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

we again can interpret the maximum likelihood result

$$\mathbf{y} \approx \boldsymbol{\Phi}\boldsymbol{\theta}_{\mathrm{ML}},$$

$$\theta_{\mathrm{ML}} = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\mathbf{y}$$

as a projection onto a K-dimensional subspace of $\mathbb{R}^N$, which is spanned by the columns of the feature matrix $\boldsymbol{\Phi}$.

LSE – MLE as Orthogonal Projection

If the feature functions ϕ_k that we use to construct the feature matrix $\mathbf{\Phi}$ are orthonormal, leads to the projection

$$\mathbf{\Phi}(\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top\mathbf{y} = \mathbf{\Phi}\mathbf{\Phi}^\top\mathbf{y} = \left(\sum_{\mathrm{k}=1}^{\mathrm{K}} \phi_k\phi_k^\top\right)\mathbf{y}$$

Thus, the maximum likelihood projection is simply the sum of the projections of $\mathbf{y}$ onto the individual basis vectors ϕ_k.

If the basis is not orthogonal $\longrightarrow$ the Gram-Schmidt process

LSE

Viewing MLE as an orthogonal projection, the NLL (negative log-likelihood) is equal (up to irrelevant constants) to the **residual sum of squares**:

$$\mathrm{RSS}(\boldsymbol{\theta}) = \frac{1}{2}\sum_{\mathrm{n}=1}^{\mathrm{N}}(y_n - \boldsymbol{\theta}^\top \mathbf{x}_n)^2 = \frac{1}{2}\left\|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\right\|^2 = \frac{1}{2}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^\top(\mathbf{X}\boldsymbol{\theta} - \mathbf{y}) \quad (17)$$

In the following, methods to optimize the (17) are discussed.

LSE – Ordinary Least Squares (OLS)

$$\nabla_{\boldsymbol{\theta}} \mathrm{RSS}(\boldsymbol{\theta}) = \mathbf{X}^\top \mathbf{X}\boldsymbol{\theta} - \mathbf{X}^\top \mathbf{y}$$

$$\xrightarrow{\text{setting the gradient to zero}} \mathbf{X}^\top \mathbf{X}\boldsymbol{\theta} = \mathbf{X}^\top \mathbf{y}$$

The above equations are known as **normal equations** (why?) and the corresponding solution $\boldsymbol{\theta}^*$ is called the **OLS** solution:

$$\boldsymbol{\theta}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

- Show that the least squares objective has a unique global minimum (show that the Hessian is positive definite).

LSE – Algorithmic Issues

For the OLS solution as

$$\boldsymbol{\theta}^* = \mathbf{X}^\dagger \mathbf{y} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y},$$

it may not be numerically possible to compute the pseudo-inverse by inverting $\mathbf{X}^\top \mathbf{X}$ since it may be singular.

A better and more general approach $\longrightarrow$ SVD

In `sklearn.linear_model.fit`, it uses `scipy.linalg.lstsq` which calls `DGELSD`, which is an SVD-based solver.
But ...

LSE – Algorithmic Issues

If $\mathbf{X}$ is tall and skinny ($N \gg D$) $\longrightarrow$ QR decomposition

Let $\mathbf{X} = \mathbf{QR}$, where $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$.

Using QR decomposition, we rewrite solving the linear system $\mathbf{X}\boldsymbol{\theta} = \mathbf{y}$ that minimizes $\|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|^2$ (If $N = D$ and $\mathbf{X}$ is full rank, the equations have unique solution and the error will be 0) as

$$
\begin{aligned}
(\mathbf{QR})\boldsymbol{\theta} &= \mathbf{y} \\
\mathbf{Q}^\top \mathbf{QR}\boldsymbol{\theta} &= \mathbf{Q}^\top \mathbf{y} \\
\boldsymbol{\theta} &= \mathbf{R}^{-1}(\mathbf{Q}^\top \mathbf{y})
\end{aligned}
$$

Thanks to this approach: backsubstitution instead of matrix inversion

LSE – Algorithmic Issues

SVD and QR are direct methods based on matrix decomposition.

Alternative approach: iterative solvers such as

- **conjugate gradient** method (assumes $\mathbf{X}$ is symmetric positive definite)
- **GMRES** (generalized minimal residual method) that works for general $\mathbf{X}$ (implemented by `sparse.linalg.gmres` in SciPy)

These are well-suited to problems where $\mathbf{X}$ is sparse or structured.

Weighted LSE

When we want to associate a weight with each example.

An instance: **heteroskedastic regression** (the variance depends on the input):

$$p(y \mid \mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y \mid \boldsymbol{\theta}^\top \mathbf{X}, \sigma^2(\mathbf{x})) = \frac{1}{\sqrt{2\pi\sigma^2(\mathbf{x})}} \exp\left(-\frac{1}{2\sigma^2(\mathbf{x})}(y - \boldsymbol{\theta}^\top \mathbf{x})^2\right)$$

$$\implies p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y} \mid \mathbf{X}\boldsymbol{\theta}, \boldsymbol{\Lambda}^{-1})$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(1/\sigma^2(\mathbf{x}_n))$.

One can show that the MLE is given by

$$\boldsymbol{\theta}^* = (\mathbf{X}^\top \boldsymbol{\Lambda} \mathbf{X})^{-1} \mathbf{X}^\top \boldsymbol{\Lambda} \mathbf{y}$$

which is known as the **weighted least squares** estimate.

Table of Contents

Measuring Goodness of Fit

Goodness of Fit is about how well a regression model fits the data.

- **Residual plots**
- **Prediction accuracy and R^2**
  - RSS
  - RMSE
  - R^2

Measuring Goodness of Fit – Residual Plots

For 1d inputs,
Check the *reasonableness* of the model by plotting the *residuals*,

$$r_n = y_n - y_n^*$$

vs the input x_n.

The model assumes that

- The residuals have a $\mathcal{N}(0, \sigma^2)$ distribution

Thus, the residual plots are of points more or less equally above and below the horizontal line at 0 with no obvious trends.

Measuring Goodness of Fit – Prediction accuracy and R^2

Rather than plotting against x_n, measure the quality of a regression's model predictions by plotting y_n^* vs y_n.

To assess the fit quantitatively using the **residual sum of squares**:

$$\mathrm{RSS}(\boldsymbol{\theta}) = \sum_{\mathrm{n}=1}^{\mathrm{N}} (y_n - \boldsymbol{\theta}^\top \mathbf{x}_n)^2$$

Another measure, **root mean squared error**:

$$\mathrm{RMSE}(\boldsymbol{\theta}) \triangleq \sqrt{\frac{1}{\mathrm{N}} \mathrm{RSS}(\boldsymbol{\theta})}$$

A more interpretable measure, **coefficient of determination**:

$$\mathrm{R}^2 \triangleq 1 - \frac{\sum_{\mathrm{n}=1}^{\mathrm{N}} (y_n^* - y_n)^2}{\sum_{\mathrm{n}=1}^{\mathrm{N}} (\bar{y} - y_n)^2} = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}}$$

Measuring Goodness of Fit – Prediction accuracy and R^2

Coefficient of Determination R^2

$$\mathrm{R}^2 \triangleq 1 - \frac{\sum_{\mathrm{n}=1}^{\mathrm{N}} (y_n^* - y_n)^2}{\sum_{\mathrm{n}=1}^{\mathrm{N}} (\bar{y} - y_n)^2} = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}}$$

where

- $\bar{y} = \frac{1}{\mathrm{N}} \sum_{\mathrm{n}=1}^{\mathrm{N}} y_n$ is the empirical mean of the response
- $\mathrm{RSS} = \frac{1}{\mathrm{N}} \sum_{\mathrm{n}=1}^{\mathrm{N}} (y_n - y_n^*)^2$ is the residual sum of squares
- $\mathrm{TSS} = \frac{1}{\mathrm{N}} \sum_{\mathrm{n}=1}^{\mathrm{N}} (y_n - \bar{y})^2$ is the total sum of squares

Note that

- R^2 measures *the variance in the prediction* relative to a simple constant prediction of $y_n^* = \bar{y}$
- $0 \leq \mathrm{R}^2 \leq 1$ (why?)
  - larger values $\longrightarrow$ a greater reduction in variance $\longrightarrow$ better fit

Overfitting

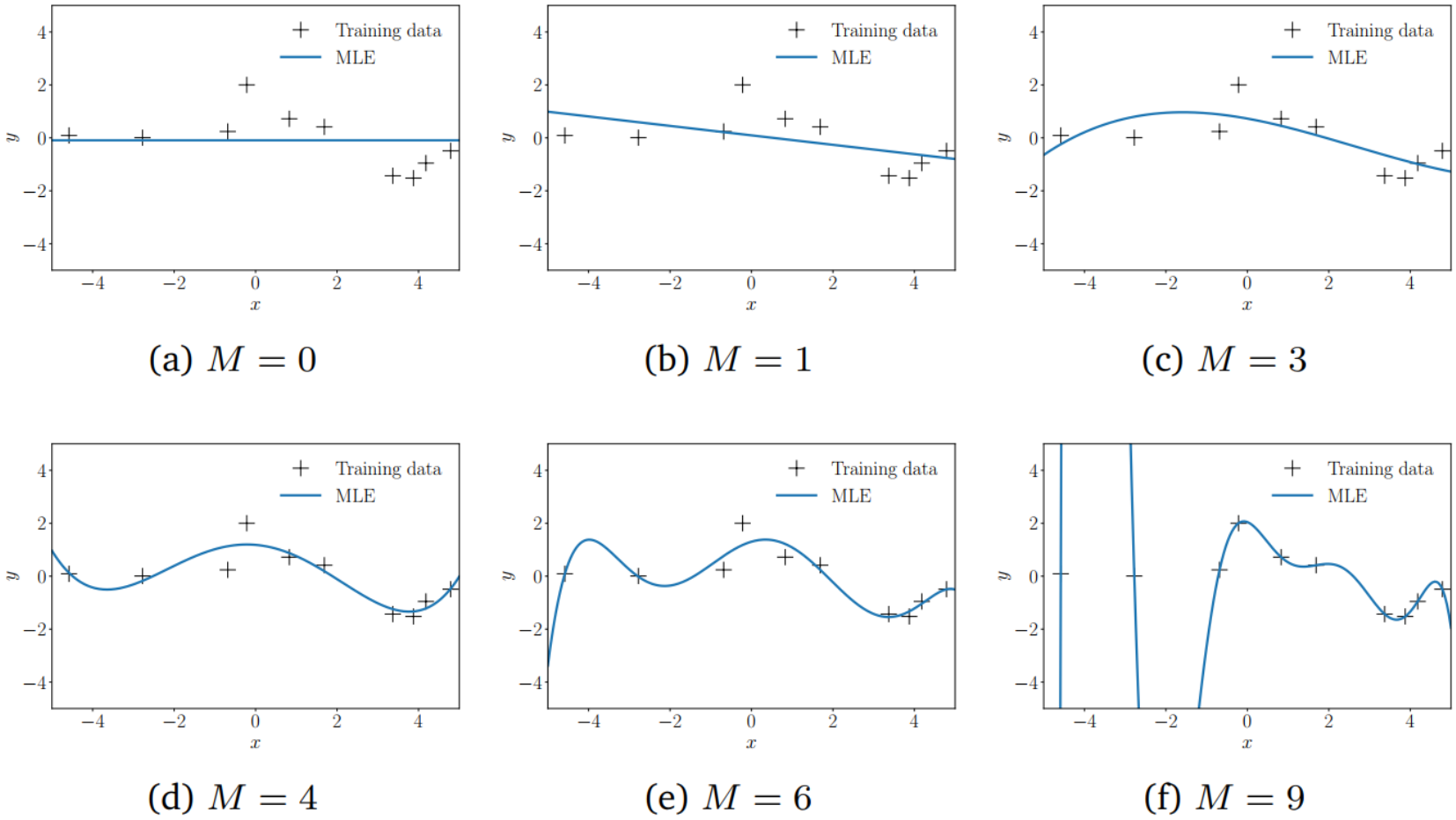
For **model selection**,

- Use RMSE (or NLL) $\longrightarrow$ find the best degree of the polynomial M that minimizes the objective
- Brute-force search $\longrightarrow$ enumerate all (reasonable) values of M
- For a training set of size N it is sufficient to test $0 \leq \mathrm{M} \leq \mathrm{N} - 1$
  - For $\mathrm{M} < \mathrm{N}$, the maximum likelihood estimator is unique.
  - For $\mathrm{M} \geq \mathrm{N}$, infinitely many possible maximum likelihood estimators.
  - The case of $\mathrm{M} = \mathrm{N} - 1$ is extreme.

Note that

- $\mathrm{M} = \mathrm{N} - 1$ is extreme in the sense that otherwise the null space of the corresponding linear system would be non-trivial $\longrightarrow$ infinitely many optimal solutions to the linear regression problem.
- For $\mathrm{M} \geq \mathrm{N}$,
  more params than data points $\longrightarrow$ undetermined system $\longrightarrow$ the $\boldsymbol{\Phi}^\top \boldsymbol{\Phi}$ in slide 34 is no longer invertible.

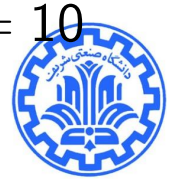
Overfitting



(a) $M = 0$ (b) $M = 1$ (c) $M = 3$

(d) $M = 4$ (e) $M = 6$ (f) $M = 9$

Figure: Maximum likelihood fits for different polynomial degrees M with $\mathrm{N} = 10$ observations



Overfitting

In the figure in previous slide,

- The polynomials of low degree (e.g., constants ($\mathrm{M} = 0$) or linear ($\mathrm{M} = 1$)) fit the data poorly $\longrightarrow$ poor representations of the true underlying function
- For degrees $\mathrm{M} = 3, \ldots, 6$, the fit looks plausible and smoothly interpolate the data
- For higher degrees, they fit the data better and better
- For the extreme case $\mathrm{M} = \mathrm{N} - 1 = 9$, the function will pass through every single data point

BUT, the high-degree polynomials (a) oscillate wildly and (b) are a poor representation of the underlying function that generated the data.

That's *overfitting*.

Overfitting

Let's test!
To gain quantitative insight into the dependence of the generalization performance on the polynomials of degree M

- A separate test set of 200 data points generated using exactly the same procedure used to generate the training set
- a linear grid of 200 points in the interval of $[-5, 5]$ as test inputs
- Evaluate the RMSE for both training and test data

Overfitting

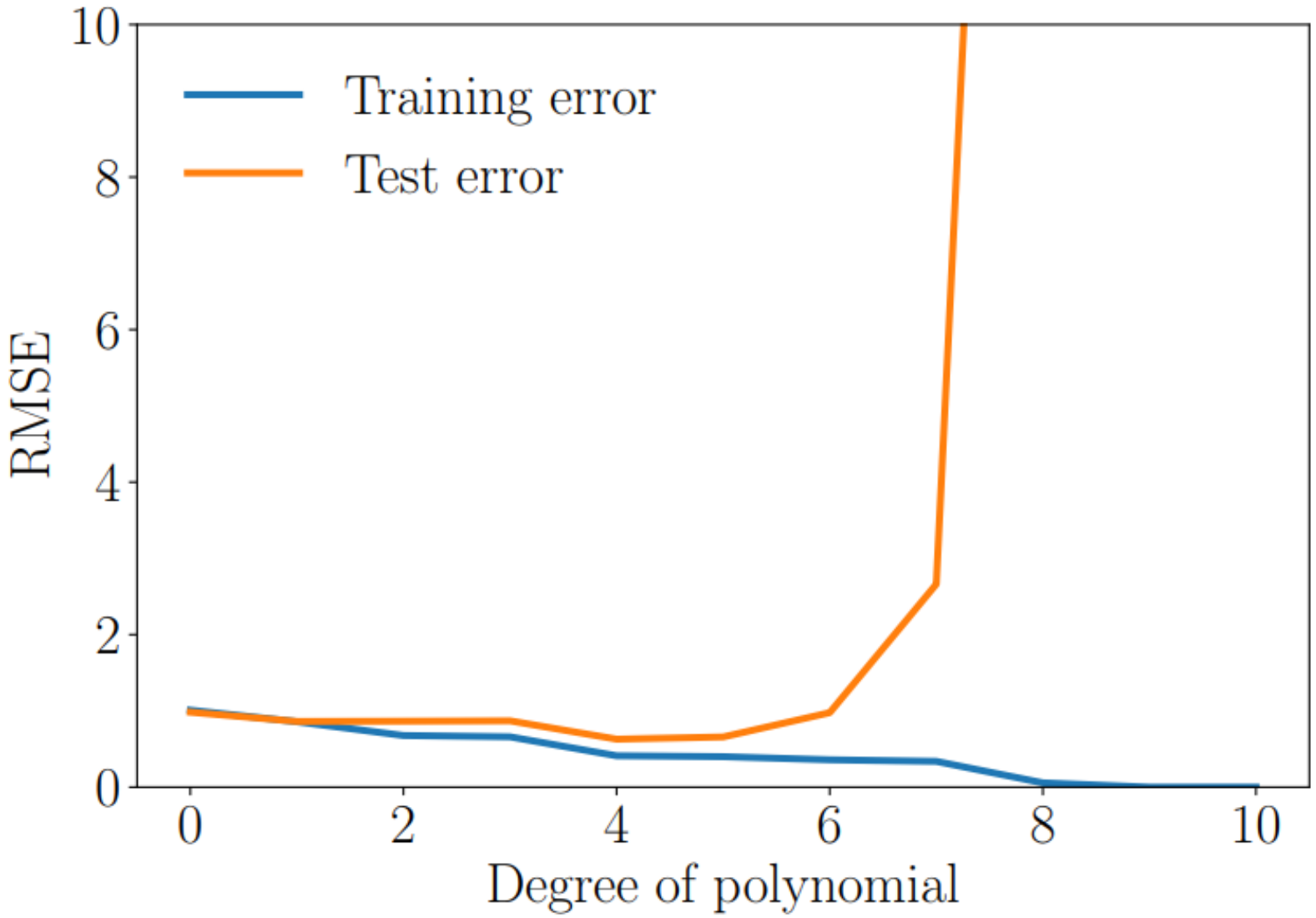


Figure: Training and test errors

Overfitting

Investigating the "training and test errors" figure,

- Test error
  - is a qualitive measure of the generalization properties of the corresponding polynomial
  - initially decreases
  - is relatively low for fourth-order polynomials ans stays relatively constant up to degree 5
  - increases significantly from degree 6 onward $\longrightarrow$ high-order polynomials have very bad generalization
- Training error never increases with the increase in the degree of the polynomial
- The best generalization $\longrightarrow$ the point of the smallest test error $\longrightarrow$ $\mathrm{M} = 4$