TECHNISCHE UNIVERSITÄT DARMSTADT

Statistical Machine Learning

Exercise Session 07: HW3 Walkthrough and Q&A

Marcus Rohrbach & Simone Schaub-Meyer
Department of Computer Science
TU Darmstadt

Summer Term 2025

Today's Objectives

1. Organizational Aspects
2. HW3 Partial Walkthrough
3. Q&A

Outline

1. Organizational Aspects

2. HW3 Partial Walkthrough

3. Q&A

Organizational Aspects

- **HW 3 Due Date:**
  - July 6th, 23:59 PM
- **HW 4 Publication Date:**
  - July 9th, after lecture
- **Next week's session** will be dedicated to:
  - HW3 Walkthrough Part 2 and Q&A
- **Attendance Testat**
  - Obligatory except for health or visa issues

Oral Examinations (Testat)

- **Location:** Gebäude S4|22, Landwehrstr. 50A
  - Tue, 08.07.2025, 11:00–15:00 – rooms 2, **5**, 6
  - Wed, 09.07.2025, 09:00–12:00 – rooms 2, 6
- **Slot selection opens Mon, 07.07.2025** – first come, first served.
- If you cannot participate, you will have to notify us beforehand and provide a **doctor's certificate**.

TECHNISCHE UNIVERSITÄT DARMSTADT

Outline

TECHNISCHE UNIVERSITÄT DARMSTADT

Linear Ridge Regression – Key Ideas

Model: $y_i = \Phi(x_i)^\top \mathbf{w} + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$

- **Loss function:**

$$\mathcal{L}(\mathbf{w}) = \|\Phi\mathbf{w} - \mathbf{y}\|^2 + \lambda\|\mathbf{w}\|^2$$

- **Ridge coefficient** λ: penalizes large weights to prevent overfitting and improve numerical stability.

- **Optimal solution:**

$$\hat{\mathbf{w}} = (\Phi^\top\Phi + \lambda\mathbf{I})^{-1}\Phi^\top\mathbf{y}$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Caution: From the Lecture Slides

- Lecture:

$$\mathbf{w} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^{\intercal} + \color{red}{\lambda}\color{black}\mathbf{I}\right)^{-1} \boldsymbol{\Phi}\mathbf{y}$$

- HW3 and Bishop:

$$\hat{\mathbf{w}} = (\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi} + \lambda\mathbf{I})^{-1}\boldsymbol{\Phi}^{\top}\mathbf{y}$$

Matter of Definition

- Defined in the lecture

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}, \quad \mathbf{\Phi} = \begin{bmatrix} | & & | \\ \phi(\mathbf{x}_1) & \dots & \phi(\mathbf{x}_n) \\ | & & | \end{bmatrix}$$

TECHNISCHE UNIVERSITÄT DARMSTADT

Matter of Definition

- In HW3 template we follow the Bishop's book for our regression setup:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_D \end{bmatrix}, \quad \mathbf{\Phi} = \begin{bmatrix} \phi_1(x_1) & \cdots & \phi_D(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_n) & \cdots & \phi_D(x_n) \end{bmatrix}$$

- Each row corresponds to a transformed input sample, and each column corresponds to a polynomial feature.
- The output model is: $\mathbf{y} \approx \mathbf{\Phi}\mathbf{w}$

TECHNISCHE UNIVERSITÄT DARMSTADT

Polynomial Features – Still Linear Models

Polynomial regression: Expand inputs using polynomial basis $\Phi(x) = [1, x, x^2, \ldots, x^d]$

- We call this *linear regression* because it's still linear in the parameters $\mathbf{w}$
- In our template, polynomial features and the bias term are handled separately:
  - `polynomial_features(X, degree)` expands the inputs as powers of x
  - `add_bias(X)` appends a constant column for the intercept term
- Increasing polynomial degree increases model flexibility but may lead to overfitting
- Use RMSE to evaluate generalization on train/test data

Bayesian Linear Regression – Core Concepts and Template Structure

TECHNISCHE UNIVERSITÄT DARMSTADT

Prior: $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \lambda^{-1}\mathbf{I})$

Posterior:

$$p(\mathbf{w} \mid \mathbf{X}, \mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Lambda}_n^{-1}) \quad \text{with} \quad \boldsymbol{\mu}_n = \boldsymbol{\Lambda}_n^{-1}\mathbf{X}^\top\mathbf{y}, \quad \boldsymbol{\Lambda}_n = \sigma^{-2}\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}$$

Predictive distribution:

$$p(y_* \mid x_*) = \mathcal{N}(x_*^\top \boldsymbol{\mu}_n, \sigma^2 + x_*^\top \boldsymbol{\Lambda}_n^{-1} x_*)$$

Template expectations:

- `fit()` method:
  - Computes $\boldsymbol{\Lambda}_n$, $\boldsymbol{\Lambda}_n^{-1}$, and $\boldsymbol{\mu}_n$
  - Stores bias flag and training data
- `predict()` method:
  - Returns predictive mean and std as derived above

Squared Exponential Features – Interpretation

Feature design:

$$\Phi_{ij} = \exp\left(-\frac{1}{2}\beta(x_i - \alpha_j)^2\right)$$

- α_j = center of basis function
- β = precision (inverse variance) of basis
- These form localized Gaussian basis functions

Model remains linear in weights $\Rightarrow$ Bayesian linear regression still applies.

TECHNISCHE UNIVERSITÄT DARMSTADT

Outline