MACHINE LEARNING: ADVANCED TECHNIQUES

Introductory Information

Institute for Machine Learning

JOHANNES KEPLER
UNIVERSITY LINZ

Institute for
Machine Learning

Contact

LVA Head: Johannnes Brandstetter

Institute for Machine Learning
Johannes Kepler University
Altenberger Str. 69
A-4040 Linz

Institute Homepage

Copyright statement:

Name of the lecture

- This a newly designed course, which was previously named **Theoretical Concepts**.
- The lecture / exercise of **Theoretical Techniques** in the SS2024 was already adjusted to match modern trends in machine learning.
- Thus, this is the third time the topics are taught, but this time again with several adjustments compared to the SS2024 and SS2025 course.

Modus

- Weekly lectures
- Always on Tuesday at 12:00 o'clock in HS 5
- First lecture: October 7, 2025
- No physical presence is required, neither for the lecture itself nor for the exams!

Evaluation

- There will be **final exams**
- The exams will take place **in persona**
- Duration: 60/90 minutes (will be announced)
- Topics: everything covered in lectures.
- In order to pass the course, more than 50% of all points on the exam are required.
- Exam and retry exam are scheduled for January 27 (13:45) and March 17 (12:00), respectively.

Communication

- Use the discussion forum (on the Moodle page of the lecture) for content related questions that may be of interest to your colleagues. And feel free to help your colleagues in the forum by answering their questions
- Additionally, there is an announcement forum for lecture related announcements.

Material

- Slides and lecture notes on Moodle
- Further sources will also be linked and announced on Moodle

Topics planned for the lecture

UNIT 1: Estimation Theory
UNIT 2: Bayes Techniques, pPCA, VAEs
UNIT 3: PDEs, Neural ODEs
UNIT 4: Diffusion models
UNIT 5: Flow matching
UNIT 6: Implicit neural representation / Neural fields
UNIT 7: Statistical Learning Theory

DISCLAIMER: This lecture is built up around the topics of generative modeling and implicit neural representation. Both topics heavily rely on the understanding of PDEs / Bayes modeling, which are introduced separately. However, it is recommended to get acquainted with PDEs / Bayes modeling if they are completely new to you.

Recap: Concepts from probability theory

- Definition of probability / probability space
- Discrete and continuous random variables
- Examples of distributions:
    1. Discrete: Bernoulli, Binomial, Poisson
    2. Continuous: Normal, Laplace, Uniform
- Associated cumulative distributions
- Conditional probability, dependence/independence of events/random variables
- Bayes Theorem, law of total probability
- Moments of random variables: expectation, variance, etc.
- Central limit theorem
- Jensen's inequality, concentration inequalities

Recap: Concepts from linear algebra

- Vector spaces, normed spaces, inner product spaces
- Linear dependence, independence, bases
- Orthogonality, angles, Cauchy-Schwartz inequality
- Linear maps and matrices, matrix operations (add, multiply, transpose)
- Special types of matrices: orthogonal, symmetric, positive definite
- Linear equations, rank, invertibility of matrices
- Determinants
- Eigenvalues and Eigenvectors, diagonalizable matrices
- Singular value decomposition

Recap: Concepts from calculus

- Limits and continuity in one and more dimensions
- Differentiation in one and more dimensions
- Differentiation with respect to vectors and matrices
- Taylor series/expansions
- Convexity
- Integration in one and more dimension (Riemann integration)
- Transformation of variables
- Basics of Hilbert spaces

Recap: Dataset

- One object is represented as **feature vector** of length d:
$$\boldsymbol{x} = (x^{(1)}, \dots, x^{(d)})^T$$
- Dataset consists of l objects with feature vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_l$
- Supervised ML: Target value $y_i \in \mathbb{R}$ for each sample $\boldsymbol{x}_i$
- All target values $\rightarrow$ **target/label vector**: $\boldsymbol{y} = (y_1, \dots, y_l)^T$
- Often dataset is summarized as **data matrix**:

$$\mathbf{Z} = \begin{pmatrix} \mathbf{X} \\ \boldsymbol{y}^T \end{pmatrix} = \begin{pmatrix} x_1^{(1)} & \dots & x_l^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(d)} & \dots & x_l^{(d)} \\ y_1 & \dots & y_l \end{pmatrix},$$

$$\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i)^T$$

- We define the training data as: $\{\boldsymbol{z}\} = \{\boldsymbol{z}_1, \dots, \boldsymbol{z}_l\}$

Recap: Unsupervised learning

- No labels
- Assumption that data are generated from a **parametrized distribution** $p(\boldsymbol{x}; \boldsymbol{w})$:
  - Tools and notions how to estimate $\boldsymbol{w}$ accurately based on the data $\mathbf{X}$ $\rightarrow$ **Estimation Theory**
- **Deep generative modeling**: Learn to sample from the data distribution without explicitly knowing the data distribution

Recap: Model and loss function

- How do we get the “best” model (in the supervised learning setup)?
    1. How does our model perform on training data? $\rightarrow$ **Loss function**
    2. How will the model perform on (unseen) future data? (i.e. how well will it generalize?) $\rightarrow$ **Generalization error/risk**
- Assume we have a model $g(\boldsymbol{x}; \boldsymbol{w})$, parameterized by $\boldsymbol{w}$.
- The output of the model should be as close as possible to the true target value y.

Recap: Model and loss function

- How do we get the “best” model (in the supervised learning setup)?
    1. How does our model perform on training data? $\rightarrow$ **Loss function**
    2. How will the model perform on (unseen) future data? (i.e. how well will it generalize?) $\rightarrow$ **Generalization error/risk**
- Assume we have a model $g(\boldsymbol{x}; \boldsymbol{w})$, parameterized by $\boldsymbol{w}$.
- The output of the model should be as close as possible to the true target value y.
- We use a **loss function**

$$L(y, g(\boldsymbol{x}; \boldsymbol{w}))$$

to measure how close our prediction is to the true target.

Recap: Examples of loss functions

Zero-one loss: $L_{\mathbf{zo}}(y, g(\boldsymbol{x}; \boldsymbol{w})) = \begin{cases} 0 & y = g(\boldsymbol{x}; \boldsymbol{w}) \\ 1 & y \neq g(\boldsymbol{x}; \boldsymbol{w}) \end{cases}$

Cross entropy loss for M classes:

$$L_{\mathbf{ce}}(y, g(\boldsymbol{x}; \boldsymbol{w})) = -\sum_{c=1}^{M} y_{p,c} \log(g(\boldsymbol{x}; \boldsymbol{w})_{p,c})$$

$y_{p,c}$: $0/1$ when class c is (in)correct for prediction p

$g(\boldsymbol{x}; \boldsymbol{w})_{p,c}$: pred. probability p that output is of class c

Binary cross entropy loss:

$$L_{\mathbf{ce}}(y, g(\boldsymbol{x}; \boldsymbol{w})) = -y \ln g(\boldsymbol{x}; \boldsymbol{w}) - (1 - y) \ln(1 - g(\boldsymbol{x}; \boldsymbol{w}))$$

Quadratic loss: $L_{\mathbf{q}}(y, g(\boldsymbol{x}; \boldsymbol{w})) = (y - g(\boldsymbol{x}; \boldsymbol{w}))^2$

Many other loss functions available with different justifications.

Recap: Generalization error, Empirical Risk Minimization

- **Joint density** of data distribution: $p(\boldsymbol{z}) = p(\boldsymbol{x}, y)$
- **Generalization error** or **risk** is the expected loss on future data:

$$R(g(.;\boldsymbol{w})) = \int_{\mathbf{X}} \int_{\mathbb{R}} L(y, g(\boldsymbol{x};\boldsymbol{w}))\, p(\boldsymbol{x}, y)\, dy\, d\boldsymbol{x}$$

- In practice, we hardly have any knowledge about $p(\boldsymbol{x}, y)$.
- Thus, we minimize the **empirical risk** R_{emp} on our dataset $\rightarrow$ **Empirical Risk Minimization(ERM)**:

$$R_{\text{emp}}(g(.;\boldsymbol{w}), \mathbf{Z}) = \frac{1}{l} \sum_{i=1}^{l} L(y_i, g(\boldsymbol{x}_i;\boldsymbol{w}))$$

Recap: Generalization error, Empirical Risk Minimization

- **Joint density** of data distribution: $p(\boldsymbol{z}) = p(\boldsymbol{x}, y)$
- **Generalization error** or **risk** is the expected loss on future data:

$$R(g(.;\boldsymbol{w})) = \int_{\mathbf{X}} \int_{\mathbb{R}} L(y, g(\boldsymbol{x};\boldsymbol{w}))\, p(\boldsymbol{x}, y)\, dy\, d\boldsymbol{x}$$

- In practice, we hardly have any knowledge about $p(\boldsymbol{x}, y)$.
- Thus, we minimize the **empirical risk** R_{emp} on our dataset $\rightarrow$ **Empirical Risk Minimization(ERM)**:

$$R_{\text{emp}}(g(.;\boldsymbol{w}), \mathbf{Z}) = \frac{1}{l} \sum_{i=1}^{l} L(y_i, g(\boldsymbol{x}_i;\boldsymbol{w}))$$

- Assume that data is **i.i.d. (independent and identically distributed)**, $\rightarrow$ **law of large numbers**:)

$$R_{\mathrm{emp}}(g(.;\boldsymbol{w})) \rightarrow R(g(.;\boldsymbol{w})) \text{ for } l \rightarrow \infty$$

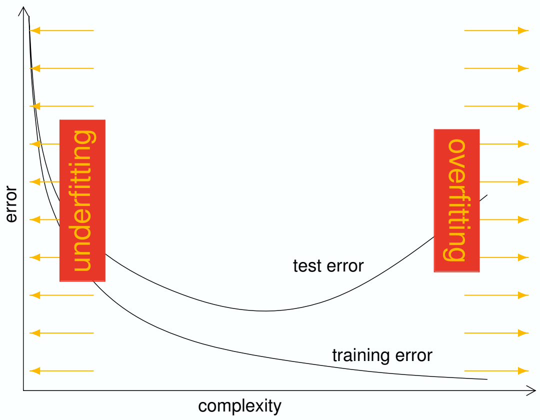
Recap: Test set method

- Assume our data samples are i.i.d.
- We can split our dataset of l samples into 2 subsets:

  Training set: a subset with m samples we perform ERM on (i.e. optimize parameters on)

  Test set: a subset with $l - m$ samples we use to estimate the risk
- Our estimate R_{emp} on the test set will show if we overfit to noise in training set

Recap: Bias-variance tradeoff



- What does “complexity” mean? What is the required setting?) $\rightarrow$ **Statistical Learning Theory**