Prediction (1)

July 14th and 15th

Supervised learning (I)

- Supervised learning is a core branch of machine learning where the goal is to learn a mapping from inputs to outputs using labeled data.
- You are given a dataset of **input-output pairs**:

$$(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)$$

- $X_i \in \mathbb{R}^p$ is a feature vector, and each Y_i (a number) is a target label or value (also called a response).
- In the tidy dataset, one of the columns is the label
- The goal is to learn a function f: $Y \approx f(X)$
- After training, we use $f(X_{n+1})$ to make predictions on **new, unseen inputs**.

Supervised learning (II)

Problem Type	Output Space	Example Tasks
Regression	Continuous (e.g. real line)	Predict house prices, temperatures
Classification	Discrete / Categorical	Predict spam/non-spam, disease/no disease

Training data: Used to fit f.
Testing data: Used to evaluate generalization (out-of-sample performance).

Concept	Description
Underfitting	Model too simple; fails to capture the pattern.
Overfitting	Model too complex; captures noise as signal.

Goal: Find a model that balances bias and variance — the classic bias-variance trade-off.

https://mlu-explain.github.io/bias-variance/

Learning goal of prediction

- Given a measurement (loss function) $\mathcal{L}$, for lots of candidates in the candidate sets $\mathcal{F}$, we want to find a $f \in \mathcal{F}$, the following values can be minimized

$$\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(Y_i, f(X_i))$$

Problem Type	Loss Function	Formula	Description
Regression	**Squared Error Loss (MSE)**	$\mathcal{L}\big(Y_i, f(X_i)\big) = \big(Y_i - f(X_i)\big)^2$	Common default.
	Absolute Error Loss(AES)	$\mathcal{L}\big(Y_i, f(X_i)\big) = \lvert Y_i - f(X_i)\rvert$	
Classification	**0-1 Loss**	$\mathcal{L}\big(Y_i, f(X_i)\big) = \mathbf{1}(Y_i = f(X_i))$	Counts misclassifications
	Cross-Entropy Loss	$\mathcal{L}\big(Y_i, f(X_i)\big) = -Y_i \log f(X_i)$	Measures distance between true and predicted distributions.

Take away:

- There are hundreds of thousands of prediction methods, mainly depending on the classes of

$$\mathcal{L}, \text{and } \mathcal{F}$$

- **Simple models** (e.g., linear regression) → high bias, low variance.
- **Complex models** (e.g., deep neural nets, decision trees without pruning) → low bias, high variance.
- Bias-variance trade off

Regression with MSE

Linear class

Algorithm	**Description**
Linear Regression (OLS) https://mlu-explain.github.io/linear-regression/	Assumes a linear relationship between input features and response.
Polynomial Regression	Extends linear regression by including polynomial terms of predictors.
Ridge Regression	Adds $\ell 2$ penalty to control coefficient size and reduce variance.
Lasso Regression	Adds $\ell 1$ penalty to encourage sparsity in coefficients.
Elastic Net	Combines $\ell 1$ and $\ell 2$ penalties.

Nonparametric class

Algorithm	Description
k-Nearest Neighbors Regression (k-NN) https://pub.towardsai.net/a-comparative-study-of-linear-and-knn-regression-a31955e6263d	Predicts based on the average of nearby neighbors.
Locally Weighted Regression (LOESS/LOWESS)	Fits local models weighted by proximity.
Spline Regression (e.g., B-splines)	Piecewise polynomials for modeling flexible curves.

Tree-Based Methods

Algorithm	Description
Decision Tree Regression https://mlu-explain.github.io/decision-tree/	Splits data recursively to model piecewise constant functions.
Random Forest Regression https://mlu-explain.github.io/random-forest/	Averaging multiple trees to reduce variance (bagging is often used).
Gradient Boosted Regression Trees (e.g., XGBoost, LightGBM, CatBoost)	Builds trees sequentially to reduce prediction error.
AdaBoost Regression	Boosts weak learners (e.g., small trees) using residuals.

Hybrid Models

Algorithm	Description
Principal Component Regression (PCR)	Applies PCA to features, then fits a linear model.
Partial Least Squares (PLS)	Projects predictors to latent space maximizing covariance with response.
Ensemble Regression (Stacking and Bagging)	Combines predictions from multiple regressors.

Other classes including Bayes regression and deep neural networks (DNN) will be discussed later this week (by Prof. Chen)