Statistical Learning: Core Ideas

Goal: Learn a function to predict a response Y from features X; quantify uncertainty and generalization error.

- Population vs sample: parameters (population quantities) vs statistics (sample estimates).
- Sampling distributions and standard error (SE): variability of a statistic across samples; larger n leads to smaller SE.
- Central Limit Theorem: many statistics (e.g., mean) are approximately normal for large n.
- Models are approximations: "All models are wrong, some are useful." Use models to compute probabilities/inference.
- Universal truth: $Y = g(X) + \delta$ with irreducible error δ. We fit $f(X)$ as an approximation to unknown $g(\cdot)$.

Bias–Variance Decomposition

- Model error at X: $(Y - \hat{f}(X))^2$ decomposes (in expectation) into bias^2 + variance + irreducible error.
- Flexibility tradeoff: more flexible models reduce bias but increase variance (overfitting risk), and vice versa.
- Larger n reduces variance of fitted models; overfitting risk is higher when n is small.

Linear Regression Essentials

Simple linear regression assumes

$$Y = \beta_0 + \beta_1 X + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2).$$

- β_1: slope (change in mean Y per unit X); β_0: intercept (mean Y at X=0).
- Least squares (LS) estimates minimize SSE $\sum_i (y_i - (\beta_0 + \beta_1 x_i))^2$.
- Residual diagnostics check linearity, constant variance, normality, independence.

Multiple linear regression

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2).$$

- Interpret β_j as partial effect of X_j holding other variables fixed.
- Challenges: multicollinearity, model uncertainty, limited visualization when $p > 2$.

Measuring Prediction Error

In-sample error (training): sample MSE

$$\text{sMSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2.$$

Out-of-sample error (future): mean squared prediction error (MSPE) on independent data (x_i^*, y_i^*),

$$\text{MSPE} = \frac{1}{n^*} \sum_{i=1}^{n^*} (y_i^* - \hat{f}(x_i^*))^2, \quad \text{EMSPE} = \mathbb{E}[(Y - \hat{Y})^2].$$

- sMSE is optimistic for complex models (re-uses training data). Use independent validation to estimate EMSPE.

Data splitting

- Train/Validation/Test (e.g., 70/15/15). Repeat splits to reduce variability; beware error leakage.

Cross-validation (CV)

- V-fold CV: split into V folds; train on V−1 folds, validate on held-out fold; average MSPE over folds.
- LOOCV: V=n. Typically choose V=5 or 10 balancing bias/variance and compute cost.
- Repeated CV: repeat random fold partitions and average to stabilize estimates.

Bootstrap (out-of-bag)

- Resample with replacement; use out-of-bag observations as validation; repeat R times and average.

Feature Engineering in Linear Models

Categorical variables

- One-hot (indicator) encoding for a factor with Q levels: include Q−1 dummies; dropped level is baseline.
- Model: $f(X) = \beta_0 + \sum_{Q=2}^{Q} \beta_Q \mathbb{I}(X{=}Q)$. Then $\beta_Q = \mu_Q - \mu_1$.

Transformations

- Polynomials $X, X^2, \ldots$; $\log X$, $\sqrt{X}$, $1/X$; choose $h(X)$ to better linearize relation with Y.

Interactions

- Cross-product $X_1 X_2$ allows slope of one variable to depend on the other: $f = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2$.

Variable Selection — Classical

Why select? Removing useless variables reduces variance (parsimony); adding variables adds variance and may reduce bias.

- All subsets: evaluate best model for each size k (SSE-minimizing); choose size by a criterion.
- Stepwise: forward (add), backward (remove), or hybrid; compare neighboring models to reduce search.

Information criteria (for ML/LS fits)

$$\text{IC}(w) = n \log(\text{sMSE}) + wk, \quad \text{with } w{=}2 \text{ (AIC)},\ w{=}\log n \text{ (BIC)},$$

select model with smallest IC; BIC penalizes complexity more than AIC.

Variable Selection — Modern (Shrinkage)

Ridge regression (L2 penalty)

$$\min_{\beta_0, \beta} \sum_{i=1}^{n} (y_i - \beta_0 - x_i^\top \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2.$$

- Shrinks coefficients toward 0 (never exactly 0); reduces variance; tune λ (e.g., GCV or CV).

LASSO (L1 penalty)

$$\min_{\beta_0, \beta} \sum_{i=1}^{n} (y_i - \beta_0 - x_i^\top \beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j|.$$

- Performs variable selection (some $\hat{\beta}_j{=}0$) and shrinkage; tune λ via CV; 1SE rule yields sparser models.
- Works well with high variance settings: multicollinearity, large p, small n.

Dimension Reduction

PCA (on standardized X): find orthogonal directions $Z_j = \sum_k \phi_{jk} X_k$ that explain decreasing variance.

- Scree/cumulative variance plots guide number of components M.

PCR: regress Y on first M PCs: $Y = \theta_0 + \sum_{m=1}^{M} \theta_m Z_m$. Treat M as tuning parameter via CV.

PLS: like PCR but learns components using correlation with Y; often needs fewer components; choose M via CV.

Step Functions (Piecewise-Constant Models)

Indicator functions for regions

- Choose cutpoints $c_1 < \cdots < c_K$; define region indicators $C_0 = \mathbb{I}(X{<}c_1)$, $C_1 = \mathbb{I}(c_1 \le X{<}c_2)$, .., $C_K = \mathbb{I}(X \ge c_K)$.
- Regression on $\{C_1, \ldots, C_K\}$ (drop one) yields a step function in X; extend via cross-products for multivariate grids.
- Cutpoints from domain knowledge or quantiles; do not use Y for cutpoint selection unless using trees.

Use

- Step functions are simple and form the basis of trees/ensembles; can approximate complex shapes via many regions.

Resampling for Model Comparison

- Use repeated splits or V-fold CV/Bootstrap to estimate EMSPE and compare multiple models.
- Visualize distributions (boxplots); relative MSPE: scale by split-wise minimum to compare stability across splits.

Key Checks and Pitfalls

- Avoid leakage: any tuning/feature engineering must be done inside training folds only.
- Parsimony: prefer simpler models when performance is similar (interpretability, cost of measurement).
- Variability: selected model (variables and size) varies across splits; treat the chosen model as an estimate.