Decision theory and supervised learning

M2 Stat SD 1

Our objective is to predict a response variable $Y \in \mathcal{Y}$ based on an observed set of covariates $X \in \mathcal{X}$ through a prediction function $f : \mathcal{X} \to \mathcal{Y}$. Given a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, we seek to identify a function f that optimizes some performance criterion — equivalently, one that minimizes prediction error on future observations (X, Y). Crucially, while we observe the covariate X for new instances, the corresponding response Y remains unobserved and must be predicted.

We formalize this setting by modeling our dataset through independent and identically distributed random pairs $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$, drawn from an unknown joint distribution p_{XY} on $\mathcal{X} \times \mathcal{Y}$. Since X is treated as random, we operate within the **random design** framework. We denote the marginal distribution of X as p_X, and when context permits, we simplify notation by writing p in place of p_{XY} or p_X.

Formally, a **machine learning algorithm** is a mapping A that takes a dataset $\mathcal{D}$ as input and produces a prediction function $f : \mathcal{X} \to \mathcal{Y}$ as output. We refer to f as a **predictor**, which may itself embody algorithmic components.

1 Decision-Theoretic Framework

To evaluate prediction quality, we introduce a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$, where $\ell(y, z)$ quantifies the penalty incurred when predicting z while the true value is y. The **expected risk** (or **generalization error**) of a predictor f is then given by

$$R(f) = \mathbb{E}\left[\ell(Y, f(X))\right] = \int_{\mathcal{X}\times\mathcal{Y}} \ell(y, f(x)) dp(x, y).$$

Since this risk functional depends on the unknown true distribution $p = p_{XY}$, we occasionally write $R_p(f)$ to emphasize this dependence explicitly.

For a given covariate value x, the **conditional risk** of making decision z is defined as

$$r(z, x) = \mathbb{E}\left[\ell(Y, z) \mid X = x\right] = \int_{\mathcal{Y}} \ell(y, z) p(y|x) dy.$$

Exercise 1.1

1. Establish the relationship between expected risk and conditional risk.
2. Consider a linear regression setting where $\mathcal{X} = \mathbb{R}$ and $\mathcal{Y} = \mathbb{R}$. Suppose the data-generating process follows
$$Y = \alpha + \beta X + \varepsilon,$$
where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ is Gaussian noise independent of X. This implies $[Y \mid X = x] \sim \mathcal{N}(\alpha + \beta x, \sigma^2)$. Derive the conditional risk for the linear predictor $f(x) = a + bx$, and subsequently compute its expected risk.

To bridge theory and practice, we introduce the **empirical risk** of a predictor f:

$$\widehat{R_n}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i)).$$

When we treat the training data as random draws (X_i, Y_i), the empirical risk becomes a random quantity, which we denote $\widehat{R}_n(f)$ to emphasize its stochastic nature.

Exercise 1.2 Compute $\mathbb{E}[\widehat{R_n}(f)]$ and interpret this result.

The **optimal predictor** minimizes expected risk over all possible prediction functions. We call $f^\star$ a **Bayes predictor** (or **Bayes classifier** in classification contexts) if it achieves the pointwise minimum:

$$f^\star(x) \in \operatorname*{argmin}_{z\in\mathcal{Y}} r(z, x) \quad \text{for all } x \in \mathcal{X}.$$

Remarkably, while multiple Bayes predictors may exist, they all achieve the same minimal expected risk, termed the **Bayes risk**:

$$R^\star = \mathbb{E}\left[\min_{z\in\mathcal{Y}} r(z, X)\right].$$

The **excess risk** of any predictor f is defined as $R(f) - R^\star$, quantifying the performance gap between f and the theoretically optimal Bayes predictor. This non-negative quantity serves as a fundamental measure of suboptimality in statistical learning theory.

Exercise 1.3

1. Prove that any Bayes predictor minimizes the expected risk among all measurable functions.
2. **Binary Classification with 0-1 Loss**: Consider $\mathcal{Y} = \{0, 1\}$ with the zero-one loss $\ell(y, z) = \mathbf{1}\{y \neq z\}$.
   - Express the expected risk of a predictor f.
   - Characterize the Bayes predictor and compute the Bayes risk.
   - Provide a formula for the excess risk of any predictor f.
3. **Regression with Squared Loss**: For $\mathcal{Y} = \mathbb{R}$ and squared error loss $\ell(y, z) = (y - z)^2$:
   - Derive the expected risk, Bayes predictor, and Bayes risk.
   - Express the excess risk decomposition.

Exercise 1.4

1. **Asymmetric Binary Classification**: Consider $\mathcal{Y} = \{0, 1\}$ with asymmetric loss structure: $\ell_{ab}(0, 0) = \ell_{ab}(1, 1) = 0$, $\ell_{ab}(0, 1) = a > 0$, $\ell_{ab}(1, 0) = b > 0$. Characterize the Bayes predictor at point x as a function of the posterior probability $\mathbb{P}(Y = 1 \mid X = x)$.
2. **Robust Regression**: Determine the Bayes predictor under absolute loss $\ell_1(y, z) = |y - z|$.
3. **Support Vector Regression**: Find the Bayes predictor for the ϵ-insensitive loss $\ell_\epsilon(y, z) = \max\{0, |y - z| - \epsilon\}$.
4. **Complementary Predictors**: For binary classification with $\mathcal{Y} = \{0, 1\}$, establish the relationship between the risks of complementary predictors f and $1 - f$.

2 Statistical Learning Theory

Statistical learning encompasses both parametric and non-parametric approaches to prediction. We begin with non-parametric methods that attempt to approximate the Bayes predictor directly, then turn to parametric methods based on empirical risk minimization.

2.1 Local Averaging Methods

Local averaging methods estimate the conditional expectation $\mathbb{E}[Y \mid X = x]$ by averaging nearby observations. These methods directly approximate the Bayes predictor for squared loss without assuming a parametric form.

2.1.1 k-Nearest Neighbors

For a query point x, the **k-nearest neighbors (k-NN)** algorithm identifies the k training points closest to x in some metric space. Let $\mathcal{N}_k(x) \subseteq \{1, \dots, n\}$ denote the indices of these k nearest neighbors.

k-NN Regression: The prediction is given by the average of the k nearest responses:

$$\hat{f}_k(x) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x)} y_i$$

k-NN Classification: For binary classification with $\mathcal{Y} = \{0, 1\}$, the prediction is:

$$\hat{f}_k(x) = \begin{cases} 1 & \text{if } \frac{1}{k} \sum_{i \in \mathcal{N}_k(x)} y_i > 1/2 \\ 0 & \text{otherwise} \end{cases}$$

Equivalently, we predict the majority class among the k nearest neighbors.

2.1.2 Nadaraya-Watson Kernel Regression

The **Nadaraya-Watson estimator** uses a scaled kernel function $K_h : \mathbb{R} \to \mathbb{R}_+$ with bandwidth $h > 0$ to weight training points based on their distance from the query point:

$$\hat{f}_h(x) = \frac{\sum_{i=1}^n K_h(x - x_i) y_i}{\sum_{i=1}^n K_h(x - x_i)}$$

The scaled kernel K_h is related to a base kernel $K : \mathbb{R}^d \to \mathbb{R}_+$ through:

$$K_h(u) = \frac{1}{h^d} K\left(\frac{u}{h}\right)$$

where $d = \dim(\mathcal{X})$ is the dimensionality of the feature space. The factor h^{-d} ensures that K_h integrates to the same value as K (typically 1 for probability kernels). The bandwidth h controls the locality of the averaging.

Common base kernel choices include: **Gaussian** $K(u) = \frac{1}{\sqrt{2\pi}} \exp(-u^2/2)$; **Epanechnikov**: $K(u) = \frac{3}{4}(1 - u^2)\mathbf{1}_{|u| \leq 1}$; **Uniform**: $K(u) = \frac{1}{2}\mathbf{1}_{|u| \leq 1}$.

Exercise 1.5 We assume that K is a probability kernel, i.e., $\int K(u)du = 1$.

1. Show that for squared loss, both k-NN and Nadaraya-Watson regression provides biased estimates of $\mathbb{E}[Y \mid X = x]$.
2. **Bias-Variance Analysis**: Discuss qualitatively how the choice of k (for k-NN) or h (for Nadaraya-Watson) affects the bias-variance tradeoff.

3. **Computational Complexity**: Compare the computational requirements of k-NN and parametric methods for both training and prediction phases.
4. **Curse of Dimensionality**: Explain why local averaging methods struggle in high-dimensional spaces.

2.2 Parametric Models and Empirical Risk Minimization

A **parametric model** consists of a family of prediction functions $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$, where Θ denotes the parameter space. Given training data, a natural approach is **empirical risk minimization (ERM)**, which seeks

$$\hat{\theta}_n \in \operatorname*{argmin}_{\theta\in\Theta} \widehat{R_n}(f_\theta) = \operatorname*{argmin}_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)).$$

The canonical example is linear regression, where $\mathcal{F} = \{f_\theta : \theta \in \mathbb{R}^d\}$ with $f_\theta(x) = \theta^\top x$.

2.3 Bias-Variance Decomposition of Excess Risk

For parametric methods, we can decompose the excess risk into two fundamental components. Let $\theta^\star \in \operatorname{argmin}_{\theta\in\Theta} R(f_\theta)$ denote the population risk minimizer within our model class. Then:

$$R(f_{\hat{\theta}_n}) - R^\star = \underbrace{R(f_{\hat{\theta}_n}) - R(f_{\theta^\star})}_{\text{estimation error}} + \underbrace{R(f_{\theta^\star}) - R^\star}_{\text{approximation error}} .$$

The **approximation error** (or **bias**) quantifies the inherent limitation of our model class — it vanishes only when the Bayes predictor lies within $\mathcal{F}$. This term depends solely on model specification, not on the learning algorithm or sample size.

The **estimation error** (or **variance**) captures the performance loss due to finite sample effects and the specific learning algorithm employed. This term typically decreases as $n \to \infty$ under appropriate regularity conditions.

Exercise 1.6 Analyze the bias-variance tradeoff: If we enlarge the function class $\mathcal{F}$, how do the approximation and estimation errors behave?

2.4 Refined Analysis of Estimation Error

We can further decompose the estimation error to understand its sources. For the population minimizer $\theta^\star$, we have:

$$\begin{aligned} R(f_{\hat{\theta}_n}) - R(f_{\theta^\star}) &= \underbrace{\left(R(f_{\hat{\theta}_n}) - \widehat{R_n}(f_{\hat{\theta}_n})\right)}_{\text{generalization gap}} + \underbrace{\left(\widehat{R_n}(f_{\hat{\theta}_n}) - \widehat{R_n}(f_{\theta^\star})\right)}_{\text{empirical optimization error}} \\ &\quad + \underbrace{\left(\widehat{R_n}(f_{\theta^\star}) - R(f_{\theta^\star})\right)}_{\text{generalization gap}} \\ &\leq 2\sup_{\theta\in\Theta}\left|R(f_\theta) - \widehat{R_n}(f_\theta)\right| + \underbrace{\widehat{R_n}(f_{\hat{\theta}_n}) - \widehat{R_n}(f_{\theta^\star})}_{\text{empirical optimization error}} . \end{aligned}$$

The first term represents the **uniform generalization gap** over the model class, while the **empirical optimization error** measures how well our algorithm approximates the population minimizer. Since $\hat{\theta}_n$ minimizes the empirical risk, the empirical optimization error is non-positive.

Exercise 1.7

1. When we consider linear models with squared loss and known σ^2, what is the empirical optimization error?
2. Generally, consider the behavior of both terms as we vary: *(i)* The complexity of the function class $\mathcal{F}$, *(ii)* The sample size n (holding the model fixed)

2.5 Regularization and Overfitting Control

To mitigate overfitting — particularly when dealing with complex model classes — we often introduce **regularization**. The **regularized empirical risk** takes the form:

$$\widehat{R}_{n,\lambda}(f_\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) + \lambda\Omega(\theta),$$

where $\Omega(\theta)$ is a **complexity penalty**, and $\lambda \geq 0$ is a **regularization parameter** controlling the bias-variance tradeoff.

Exercise 1.8 Identify several machine learning algorithms that employ regularized empirical risk minimization. Discuss the form of their regularization terms.

2.6 Data Splitting and Model Assessment

In practice, we partition available data into three distinct sets:

- **Training set** (size n): Used for parameter estimation via ERM or regularized ERM
- **Validation set** (size m): Used for hyperparameter tuning (e.g., selecting λ)
- **Test set** (size m'): Used for final performance evaluation with the chosen hyperparameters

This separation is crucial for honest assessment of generalization performance.

Exercise 1.9

1. **Overfitting Analysis**: Among the risk quantities we have defined, which increase and which decrease when a model overfits to training data?
2. **Test Risk Interpretation**: Define what you understand by "test risk" and explain how it relates to the theoretical risk quantities we have studied. What insights does test performance provide about model quality?