Contents

1 Statistical learning

Statistical learning involves tools and techniques for modeling and understanding data. It is fundamental to tasks in business analytics and spans two key goals: *inference* and *prediction*.

1.1 Inference and Prediction

- **Inference:** Understand the relationship between predictors X and response Y. For example, determining which predictors significantly affect Y and the nature of this relationship.
- **Prediction:** Focus on accurately predicting Y based on new X values. This requires estimating $f(X)$ in the model:

  $$Y = f(X) + \epsilon,$$

  where ϵ is the irreducible error, assumed to have mean zero and constant variance.

1.2 Bias-Variance Tradeoff

The expected test mean squared error (MSE) for a prediction model is given by:

$$E\big[(Y - \hat{f}(X))^2\big] = Bias^2(\hat{f}(X)) + Var(\hat{f}(X)) + \sigma^2,$$

where:

- **Bias:** Error introduced by approximating a complex reality with a simpler model.
- **Variance:** Error due to model sensitivity to training data variations.
- σ^2: Irreducible error.

This tradeoff guides the selection of model complexity.

2 Supervised vs. Unsupervised Learning

2.1 Supervised Learning

Supervised learning methods involve a set of predictors X and a response Y. Examples include:

- **Regression:** Predicting quantitative responses.
- **Classification:** Assigning categories to observations.

2.2 Unsupervised Learning

Unsupervised learning methods analyze data without labeled responses, identifying patterns or structures. Examples:

- **Clustering:** Grouping observations into clusters.
- **Dimensionality Reduction:** Simplifying data representations.

3 Model Accuracy

Model accuracy is assessed using metrics like Mean Squared Error (MSE) in regression:

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

In classification, accuracy is measured by the error rate:

$$ErrorRate = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i),$$

where $I(\cdot)$ is an indicator function.

3.1 Choosing Models

Key considerations include:

- **Flexibility:** Flexible models fit data closely but risk overfitting.
- **Interpretability:** Simple models are easier to understand but may underfit.

Cross-validation techniques are used to estimate test MSE and select models with optimal bias-variance tradeoffs.

4 Linear Regression and K-Nearest Neighbors (KNN)

This section discusses two fundamental methods in statistical learning: linear regression and K-nearest neighbors, emphasizing their mathematical formulation, applications, and limitations.

4.1 Linear Regression

Linear regression models the relationship between a quantitative response Y and one or more predictors $X_1, X_2, \ldots, X_p$ using a linear function. The model is given by:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon,$$

where:

- $\beta_0, \beta_1, \ldots, \beta_p$ are the regression coefficients.
- ϵ is the error term, assumed to have $E[\epsilon] = 0$ and constant variance σ^2.

4.1.1 Estimating Coefficients

The coefficients $\beta_0, \beta_1, \ldots, \beta_p$ are estimated using *ordinary least squares* (OLS), minimizing the residual sum of squares (RSS):

$$RSS = \sum_{i=1}^{n} \big(y_i - \hat{y}_i\big)^2 = \sum_{i=1}^{n} \big(y_i - (\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})\big)^2.$$

4.1.2 Assessing Model Accuracy

Model accuracy can be evaluated using:

- **R-squared (R^2):** Proportion of variance in Y explained by the predictors:

  $$R^2 = 1 - \frac{RSS}{TSS},$$

  where TSS is the total sum of squares.

- **Mean Squared Error (MSE):** Measures prediction error:

  $$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

4.1.3 Limitations

Linear regression assumes:

- Linear relationship between predictors and response.
- Homoscedasticity: Constant variance of errors.
- Independence of errors.
- Normally distributed errors (for inference purposes).

Violations of these assumptions can reduce model performance.

4.2 K-Nearest Neighbors (KNN)

KNN is a non-parametric method used for both regression and classification. It predicts the response Y of a test observation x_0 by considering the K closest points in the training data.

4.2.1 Algorithm

For a test observation x_0:

1. Compute the Euclidean distance between x_0 and all training points.
2. Identify the K closest points (*neighbors*).
3. **For regression:** Predict $\hat{y}$ as the average of the responses of the K neighbors:
$$\hat{y} = \frac{1}{K} \sum_{i \in N_0} y_i,$$
where N_0 is the set of K nearest neighbors.
4. **For classification:** Assign x_0 to the most common class among its K neighbors.

4.2.2 Tradeoffs

The choice of K controls the model's flexibility:

- K small: High flexibility, low bias, but high variance (overfitting).
- K large: Low flexibility, high bias, but low variance (underfitting).

4.2.3 Comparison with Linear Regression

- Linear regression assumes a specific functional form (*parametric model*), while KNN makes no such assumption (*non-parametric*).
- KNN adapts to data complexity but requires large datasets for reliable predictions.
- Linear regression is interpretable and efficient but may struggle with non-linear relationships.

5 Classification: Maximum Likelihood Estimation and Logistic Regression

Classification involves predicting a qualitative response variable Y by associating observations X with one of K classes. Logistic regression is a widely used method for binary and multi-class classification problems.

5.1 Maximum Likelihood Estimation (MLE)

Maximum Likelihood Estimation is a fundamental method for estimating parameters of a probabilistic model. It identifies the parameter values that maximize the likelihood of the observed data.

5.1.1 Likelihood Function

Suppose we have n independent observations $(x_1, y_1), \ldots, (x_n, y_n)$, where $y_i \in \{0, 1\}$. Let $p(x_i) = P(Y = 1 \mid X = x_i)$. The likelihood function is given by:

$$L(\beta) = \prod_{i=1}^{n} p(x_i)^{y_i} \big(1 - p(x_i)\big)^{1-y_i},$$

where β represents the model parameters.

5.1.2 Log-Likelihood Function

Taking the logarithm of the likelihood function for computational simplicity:

$$\ell(\beta) = \sum_{i=1}^{n} \big[y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i))\big].$$

The MLE is obtained by maximizing $\ell(\beta)$ with respect to β.

5.2 Logistic Regression

Logistic regression models the probability that a given observation belongs to a particular class. For binary classification ($Y \in \{0, 1\}$), the model assumes:

$$P(Y = 1 \mid X = x) = \frac{\exp(\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p)}{1 + \exp(\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p)}.$$

Equivalently, the *log-odds* or *logit* of the probability is linear in the predictors:

$$\log\left(\frac{P(Y = 1 \mid X = x)}{P(Y = 0 \mid X = x)}\right) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p.$$

5.2.1 Estimating Parameters

The coefficients $\beta_0, \beta_1, \ldots, \beta_p$ are estimated by maximizing the log-likelihood $\ell(\beta)$. Optimization algorithms like Newton-Raphson are commonly used.

5.2.2 Interpretation of Coefficients

- β_j represents the change in the log-odds of $Y = 1$ for a one-unit increase in x_j, holding all other predictors constant.
- Exponentiating β_j gives the *odds ratio*:

$$\mathit{OddsRatio} = \exp(\beta_j).$$

5.3 Assessing Model Performance

To evaluate the performance of a classification model, common metrics include:

- **Confusion Matrix:** Summarizes the counts of true positives, false positives, true negatives, and false negatives.
- **Accuracy:** Fraction of correctly classified observations:

$$Accuracy = \frac{NumberofCorrectPredictions}{TotalObservations}.$$

- **ROC Curve and AUC:** The Receiver Operating Characteristic (ROC) curve plots the true positive rate against the false positive rate for various thresholds. The area under the curve (AUC) measures overall model performance.

5.4 Extensions of Logistic Regression

5.4.1 Multinomial Logistic Regression

For $K > 2$ classes, logistic regression generalizes to the multinomial case, modeling:

$$P(Y = k \mid X = x) = \frac{\exp(\beta_{0k} + \beta_{1k}x_1 + \ldots + \beta_{pk}x_p)}{\sum_{j=1}^{K} \exp(\beta_{0j} + \beta_{1j}x_1 + \ldots + \beta_{pj}x_p)}.$$

5.4.2 Regularization

Regularization techniques like *ridge regression* or *lasso* can be applied to logistic regression to prevent overfitting:

$$\ell_{regularized}(\beta) = \ell(\beta) - \lambda \sum_{j=1}^{p} |\beta_j| \quad (Lasso),$$

where λ controls the penalty for large coefficients.

5.5 Comparison to KNN

While logistic regression provides an interpretable model for classification, it assumes a specific functional form. In contrast, KNN is non-parametric and relies solely on the data, offering flexibility at the cost of interpretability and computational efficiency.

6 Resampling Methods: Cross-Validation

Resampling methods are statistical techniques used to estimate the performance of a model by repeatedly sampling from the dataset. Cross-validation (CV) is a key resampling method used to assess a model's predictive accuracy and prevent overfitting.

6.1 Overview of Resampling Methods

Resampling methods involve repeatedly drawing subsets of data and using them to train and test a model. Common resampling methods include:

- **Cross-Validation:** Splitting the dataset into training and test sets multiple times to evaluate model performance.
- **Bootstrap:** Sampling with replacement to estimate the variability of a statistic (discussed in later sections of the chapter).

6.2 Cross-Validation (CV)

Cross-validation is used to estimate the test error of a model by dividing the data into training and validation subsets. It is a crucial tool for model selection and hyperparameter tuning.

6.2.1 Validation Set Approach

In the validation set approach, the dataset is randomly split into two parts:

- **Training Set:** Used to fit the model.
- **Validation Set:** Used to assess the model's performance.

The test error is estimated as the error on the validation set. However, this method can yield high variability, as it depends heavily on the specific split.

6.2.2 k-Fold Cross-Validation

k-fold cross-validation divides the data into k roughly equal-sized folds. The process involves:

1. Splitting the dataset into k folds.
2. Iteratively using $k-1$ folds for training and the remaining fold for testing.
3. Computing the test error for each fold and averaging the results to estimate the overall test error.

The k-fold cross-validation error is given by:

$$CV_k = \frac{1}{k}\sum_{i=1}^{k} MSE_i,$$

where MSE_i is the mean squared error for the i-th fold.

6.2.3 Leave-One-Out Cross-Validation (LOOCV)

LOOCV is a special case of k-fold cross-validation where $k = n$ (number of observations). Each observation is treated as a validation set, and the model is trained on the remaining $n - 1$ observations. The LOOCV error is computed as:

$$LOOCV Error = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{f}_{-i}(x_i)\right)^2,$$

where $\hat{f}_{-i}(x_i)$ is the prediction for x_i obtained by training the model on all observations except i.

6.3 Bias-Variance Tradeoff in Cross-Validation

- **LOOCV:** Low bias but potentially high variance due to training on nearly the entire dataset.
- **k-Fold CV:** Higher bias for small k, but lower variance compared to LOOCV. $k = 5$ or $k = 10$ is often a good tradeoff.

6.4 Model Selection and Hyperparameter Tuning

Cross-validation is commonly used for selecting the best model or hyperparameters. For example:

- Comparing test errors across multiple models.
- Selecting the regularization parameter λ in ridge regression or the number of neighbors K in KNN.

6.5 Advantages and Disadvantages of Cross-Validation

- **Advantages:**
  - Provides an unbiased estimate of test error.
  - Useful for model comparison and hyperparameter selection.
- **Disadvantages:**
  - Computationally intensive, especially for large datasets or complex models.
  - LOOCV can suffer from high variance.

6.6 Comparison of Resampling Methods

- Validation set approach is simple but has high variability.
- LOOCV minimizes bias but can have high variance.
- k-fold CV balances bias and variance and is computationally efficient for moderate k.

7 Resampling Methods: The Bootstrap

The bootstrap is a powerful and flexible resampling method used to estimate the accuracy of statistical estimates, such as standard errors and confidence intervals, by resampling with replacement from the observed data.

7.1 Introduction to the Bootstrap

The bootstrap is a non-parametric approach that relies on repeatedly drawing samples (with replacement) from the observed data to create multiple bootstrap datasets. It is particularly useful when the theoretical calculation of standard errors is challenging.

7.1.1 Bootstrap Process

Given a dataset $\{x_1, x_2, \dots, x_n\}$:

1. Generate B bootstrap samples, each of size n, by sampling with replacement from the original dataset.
2. Compute the statistic of interest (e.g., mean, variance, regression coefficient) for each bootstrap sample.
3. Use the distribution of the B bootstrap estimates to infer properties such as bias, variance, or confidence intervals.

7.2 Bootstrap Estimate of Standard Error

The bootstrap can estimate the standard error of a statistic $\hat{\theta}$, such as the sample mean or regression coefficient. For B bootstrap samples, let $\hat{\theta}_1^*, \hat{\theta}_2^*, \dots, \hat{\theta}_B^*$ represent the estimates from each bootstrap sample. The bootstrap estimate of the standard error is:

$$SE(\hat{\theta}) = \sqrt{\frac{1}{B-1} \sum_{b=1}^{B} \left(\hat{\theta}_b^* - \bar{\theta}^*\right)^2},$$

where $\bar{\theta}^* = \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}_b^*$ is the mean of the bootstrap estimates.

7.3 Bootstrap Confidence Intervals

Bootstrap confidence intervals can be constructed using the empirical distribution of the bootstrap estimates:

- **Percentile Method:** Use the $\alpha/2$ and $1-\alpha/2$ percentiles of the bootstrap distribution to form a $1-\alpha$ confidence interval.

$$CI_{percentile} = \left[\hat{\theta}_{\alpha/2}^*, \hat{\theta}_{1-\alpha/2}^*\right].$$

- **Standard Normal Method:** If the statistic $\hat{\theta}$ is approximately normal, construct the interval as:

  $$CI_{normal} = \left[\hat{\theta} - z_{1-\alpha/2} \cdot SE(\hat{\theta}), \hat{\theta} + z_{1-\alpha/2} \cdot SE(\hat{\theta})\right],$$

  where $z_{1-\alpha/2}$ is the critical value from the standard normal distribution.

7.4 Applications of the Bootstrap

The bootstrap is widely used in the following scenarios:

- **Standard Error Estimation:** For statistics where theoretical formulas are difficult to derive.
- **Confidence Intervals:** Provides robust intervals when standard methods are unreliable.
- **Bias Correction:** Estimates the bias of a statistic:

  $$Bias(\hat{\theta}) = \bar{\theta}^* - \hat{\theta},$$

  where $\hat{\theta}$ is the statistic computed from the original dataset.
- **Model Validation:** Evaluate the variability of model estimates in regression or classification tasks.

7.5 Advantages and Limitations of the Bootstrap

- **Advantages:**
  - Applicable to a wide range of statistics and models.
  - Does not require complex mathematical derivations.
  - Flexible for small sample sizes.
- **Limitations:**
  - Computationally intensive, especially for large datasets or complex models.
  - Assumes the sample is representative of the population.

7.6 Comparison to Cross-Validation

While both bootstrap and cross-validation involve resampling, their purposes differ:

- **Cross-Validation:** Focuses on model evaluation and hyperparameter tuning by estimating test error.
- **Bootstrap:** Focuses on estimating the accuracy (e.g., standard error, bias) of a specific statistic.

8 Linear Model Selection Methods: Subset and Stepwise Selection

Model selection methods aim to identify the subset of predictors in a linear regression model that provides the best balance between prediction accuracy and interpretability. This section covers subset selection and stepwise selection methods.

8.1 Subset Selection

Subset selection involves identifying the best subset of predictors from a larger set.

8.1.1 Best Subset Selection

Best subset selection considers all possible subsets of the predictors and selects the one that minimizes a specified criterion, such as residual sum of squares (RSS) or maximizes R^2.

1. Fit all 2^p models (where p is the number of predictors).
2. Evaluate each model using a criterion such as:
    - **RSS:** Residual sum of squares.
    - **R^2:** Proportion of variance explained.
    - **AIC/BIC:** Information criteria penalizing for complexity.
    - **Adjusted R^2:** Accounts for the number of predictors.
3. Choose the best model based on the criterion.

8.1.2 Advantages and Limitations

- **Advantages:**
    - Evaluates all possible models, ensuring the best subset is found.
- **Limitations:**
    - Computationally expensive for large p.
    - Risk of overfitting if the evaluation metric is not properly penalized.

8.2 Stepwise Selection

Stepwise selection is a more computationally efficient alternative to best subset selection, using a step-by-step approach to add or remove predictors.

8.2.1 Forward Stepwise Selection

Starts with no predictors and iteratively adds predictors that improve the model:

1. Begin with the null model ($Y = \beta_0 + \epsilon$).
2. Add the predictor that results in the largest improvement in the chosen criterion (e.g., R^2, AIC, BIC).
3. Repeat until adding more predictors does not significantly improve the model.

8.2.2 Backward Stepwise Selection

Starts with all predictors and iteratively removes the least significant predictors:

1. Begin with the full model ($Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \epsilon$).
2. Remove the predictor with the least impact on the model based on a chosen criterion (e.g., p-value, AIC, BIC).
3. Repeat until no predictors can be removed without significantly reducing the model's performance.

8.2.3 Hybrid Stepwise Selection

Combines forward and backward selection:

- Adds predictors as in forward selection but allows for the removal of predictors at each step if they become insignificant.

8.3 Model Selection Criteria

Common criteria for model evaluation include:

- **Akaike Information Criterion (AIC):**
$$AIC = -2\ell + 2p,$$
where ℓ is the log-likelihood, and p is the number of predictors.
- **Bayesian Information Criterion (BIC):**
$$BIC = -2\ell + p\log(n),$$
where n is the number of observations.
- **Adjusted R^2:**
$$R^2_{adj} = 1 - \frac{RSS/(n-p-1)}{TSS/(n-1)}.$$

8.4 Advantages and Limitations of Stepwise Selection

- **Advantages:**
  - Computationally efficient, especially for large p.
  - Easy to implement and interpret.
- **Limitations:**
  - May not find the optimal subset.
  - Prone to overfitting if not properly regularized.
  - Sensitive to the choice of stopping criteria.

8.5 Comparison of Subset and Stepwise Selection

- **Subset Selection:** Guarantees the best model (based on the evaluation metric) but is computationally infeasible for large p.
- **Stepwise Selection:** More practical for large p, but may miss the optimal model.

9 Regularization: Ridge Regression and the LASSO

Regularization methods are used to improve the performance of linear models by penalizing large coefficients. This helps prevent overfitting, especially when dealing with high-dimensional data ($p \gg n$).

9.1 Introduction to Regularization

The goal of regularization is to constrain or shrink the coefficient estimates to reduce model complexity and enhance generalization. The general form of a regularized linear regression model is:

$$\hat{\beta} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \cdot P(\beta) \right\},$$

where:

- $\lambda \geq 0$ is the regularization parameter.
- $P(\beta)$ is the penalty term that constrains the magnitude of coefficients.

9.2 Ridge Regression

Ridge regression imposes an L_2-norm penalty on the coefficients. The optimization problem is:

$$\hat{\beta}^{ridge} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}.$$

9.2.1 Key Properties

- The L_2-penalty shrinks the coefficients toward zero but does not set them exactly to zero.
- It is particularly useful when $p > n$ or when predictors are highly correlated.

9.2.2 Ridge Regression Solution

Ridge regression has a closed-form solution:

$$\hat{\beta}^{ridge} = (X^\top X + \lambda I)^{-1} X^\top y,$$

where I is the identity matrix and λ controls the amount of regularization.

9.2.3 Choosing λ

The regularization parameter λ is typically selected using cross-validation to balance bias and variance:

- Large λ: Greater shrinkage, reduced variance, increased bias.
- Small λ: Less shrinkage, reduced bias, increased variance.

9.3 The LASSO (Least Absolute Shrinkage and Selection Operator)

The LASSO applies an L_1-norm penalty, leading to sparse solutions where some coefficients are exactly zero. The optimization problem is:

$$\hat{\beta}^{lasso} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}.$$

9.3.1 Key Properties

- The L_1-penalty forces some coefficients to be exactly zero, performing variable selection.
- LASSO is effective for sparse models where only a subset of predictors is relevant.

9.3.2 LASSO vs. Ridge Regression

- Ridge regression shrinks coefficients continuously but does not eliminate predictors.
- LASSO performs variable selection by setting some coefficients to exactly zero.
- Ridge regression is preferable when all predictors are relevant, while LASSO is suitable when many predictors are irrelevant.

9.4 Comparing Ridge and LASSO

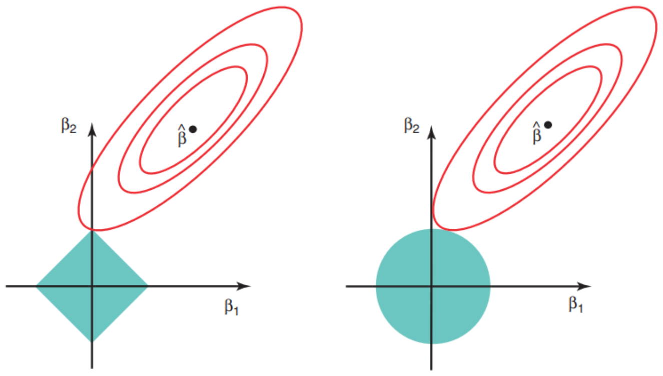

Figure 1: Constraint regions for ridge regression (L_2) and LASSO (L_1). Ridge regression minimizes within a circular region, while LASSO minimizes within a diamond-shaped region, leading to sparse solutions.

9.5 Elastic Net

Elastic Net combines the L_1- and L_2-penalties:

$$\hat{\beta}^{elastic} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \right\}.$$

This method retains the variable selection property of LASSO while addressing the limitations of highly correlated predictors.

9.6 Selecting the Regularization Parameter

- Use k-fold cross-validation to find the value of λ that minimizes the cross-validated prediction error.
- Plot the error as a function of λ (e.g., using a "regularization path").

9.7 Advantages and Limitations

- **Advantages:**
  - Reduces overfitting by controlling model complexity.
  - LASSO performs variable selection, improving interpretability.
- **Limitations:**
  - Ridge regression does not perform variable selection.

- LASSO struggles when predictors are highly correlated, as it arbitrarily selects one predictor from a group.
- Regularization requires careful tuning of λ, often computationally intensive.

10 Non-Linear Models: Regression Splines and Local Regression Methods

Non-linear models provide flexibility to capture relationships between predictors and response variables that cannot be well represented by linear models. This section focuses on regression splines and local regression methods.

10.1 Regression Splines

Regression splines are an extension of piecewise polynomial regression that ensure smoothness at the boundaries of the pieces, known as knots.

10.1.1 Piecewise Polynomials

Piecewise polynomials divide the range of a predictor X into intervals at specified *knots* and fit separate polynomial models to each interval:

$f(x) = \{ \beta_0 + \beta_1 x + \ldots + \beta_d x^d, if x \in Interval 1, \gamma_0 + \gamma_1 x + \ldots + \gamma_d x^d, if x \in Interval 2.$

However, this can result in discontinuities at the knots.

10.1.2 Definition of Splines

Splines improve piecewise polynomials by ensuring smoothness:

$$f(x) = \sum_{j=1}^{p} \beta_j B_j(x),$$

where $B_j(x)$ are basis functions that define the spline.

10.1.3 Types of Splines

- **Cubic Splines:** Use piecewise cubic polynomials that are continuous up to the second derivative at the knots.
- **Natural Splines:** Constrain the function to be linear in the boundary regions, reducing flexibility near the extremes.

10.1.4 Choosing Knots

The placement and number of knots influence the flexibility of the model. Strategies include:

- Placing knots at fixed quantiles of the data.
- Using cross-validation to select the number and position of knots.

10.1.5 Advantages and Limitations

- **Advantages:** Smooth and flexible, handles non-linear relationships well.
- **Limitations:** Requires careful selection of the number and position of knots.

10.2 Smoothing Splines

Smoothing splines balance flexibility and smoothness using a penalty term:

$$\min_f \left\{ \sum_{i=1}^{n} \big(y_i - f(x_i)\big)^2 + \lambda \int \big(f''(x)\big)^2 dx \right\},$$

where λ controls the tradeoff between fit and smoothness:

- Large λ: Produces a smoother function.
- Small λ: Produces a more flexible function.

10.3 Local Regression

Local regression is a non-parametric method that fits a regression model at each target point x_0 using only observations near x_0.

10.3.1 Local Weighted Regression

The model is fit by minimizing a weighted least squares criterion:

$$\min_{\beta_0,\beta_1} \sum_{i=1}^{n} w_i \big(y_i - \beta_0 - \beta_1 x_i\big)^2,$$

where w_i are weights assigned to each observation based on its distance from x_0. A common weighting function is:

$$w_i = K\left(\frac{|x_i - x_0|}{h}\right),$$

where h is the bandwidth and K is a kernel function (e.g., Gaussian, Epanechnikov).

10.3.2 Bandwidth Selection

The bandwidth h controls the smoothness of the local regression:

- Small h: High flexibility, risk of overfitting.
- Large h: Low flexibility, risk of underfitting.

Cross-validation is often used to select the optimal bandwidth.

10.3.3 Advantages and Limitations

- **Advantages:** Highly flexible, no need to specify a global model.
- **Limitations:** Computationally intensive, especially for large datasets; sensitive to bandwidth selection.

10.4 Comparison of Regression Splines and Local Regression

- **Regression Splines:**
  - Parametric approach.
  - Requires selection of knots but computationally efficient.
- **Local Regression:**
  - Non-parametric approach.
  - Adaptable to complex relationships but computationally intensive.

10.5 Applications and Use Cases

- **Regression Splines:** Suitable for problems requiring smoothness and global interpretability (e.g., predicting trends over time).
- **Local Regression:** Best for capturing highly localized patterns in data (e.g., spatial or temporal variability).

11 Smoothing Splines and Generalized Additive Models (GAMs)

Smoothing splines and generalized additive models (GAMs) are powerful tools for capturing non-linear relationships between predictors and the response variable while maintaining interpretability and flexibility.

11.1 Smoothing Splines

Smoothing splines are an extension of regression splines that automatically determine the level of smoothness using a penalty term.

11.1.1 Definition of Smoothing Splines

A smoothing spline is a function $f(x)$ that minimizes the following penalized residual sum of squares:

$$\min_f \left\{ \sum_{i=1}^{n} \big(y_i - f(x_i)\big)^2 + \lambda \int \big(f''(x)\big)^2 dx \right\},$$

where:

- The first term ensures a good fit to the data.
- The second term penalizes excessive curvature, controlled by the smoothing parameter λ.

11.1.2 Role of the Smoothing Parameter (λ)

- $\lambda = 0$: No penalty, resulting in an interpolating spline.
- $\lambda \to \infty$: Maximum penalty, resulting in a linear fit.
- Optimal λ: Chosen using cross-validation to balance bias and variance.

11.1.3 Advantages and Limitations

- **Advantages:**
  - Provides a smooth and flexible fit to the data.
  - Automatically balances fit and smoothness.
- **Limitations:**
  - Computationally intensive for very large datasets.
  - Requires careful selection of λ.

11.2 Generalized Additive Models (GAMs)

GAMs extend linear models by allowing for non-linear relationships between each predictor and the response while retaining additive structure.

11.2.1 Model Definition

The GAM for a response Y is given by:

$$g(E[Y]) = \beta_0 + f_1(X_1) + f_2(X_2) + \ldots + f_p(X_p),$$

where:

- $g(\cdot)$: Link function, e.g., identity for regression or logit for classification.
- $f_j(X_j)$: A non-linear, smooth function of predictor X_j, often estimated using splines.

11.2.2 Fitting GAMs

The $f_j(X_j)$ functions are estimated using techniques such as smoothing splines, local regression, or regression splines. The estimation is performed by minimizing a penalized likelihood function:

$$\ell(\beta_0, f_1, \ldots, f_p) - \lambda \sum_{j=1}^{p} \int \left(f_j''(x)\right)^2 dx.$$

11.2.3 Interpretability of GAMs

- Additive structure allows for easy interpretation of the relationship between individual predictors and the response.
- The effect of each predictor X_j on Y can be visualized by plotting $f_j(X_j)$.

11.2.4 Applications of GAMs

GAMs are useful in scenarios where:

- Non-linear relationships exist between predictors and the response.
- Interpretability of individual predictor effects is essential.

11.3 Comparison of Smoothing Splines and GAMs

- **Smoothing Splines:** Applied to a single predictor-response relationship, ensuring smoothness while fitting the data.
- **GAMs:** Extend the concept of smoothing splines to multiple predictors, allowing additive, non-linear relationships.

11.4 Advantages and Limitations of GAMs

- **Advantages:**
  - Combines flexibility with interpretability.
  - Suitable for regression and classification tasks.
- **Limitations:**
  - Assumes additivity, which may not capture interactions between predictors.
  - Computationally intensive for large datasets or complex models.

11.5 Choosing Between Models

- Use smoothing splines for simple relationships involving a single predictor.
- Use GAMs when modeling non-linear effects of multiple predictors with a focus on interpretability.

12 Tree-Based Methods: Regression and Classification Trees

Tree-based methods are versatile models used for both regression and classification tasks. These methods partition the feature space into regions and make predictions based on the mean or mode of the observations within each region.

12.1 Introduction to Decision Trees

Decision trees recursively split the predictor space into non-overlapping regions to predict a response Y. The resulting model is intuitive and interpretable.

12.2 Regression Trees

12.2.1 Definition

Regression trees predict a quantitative response by dividing the predictor space into J regions $R_1, R_2, \ldots, R_J$ and assigning the mean response value of the training observations in each region as the predicted value:

$$\hat{Y} = \sum_{j=1}^{J} \bar{y}_{R_j} I(x \in R_j),$$

where $\bar{y}_{R_j}$ is the mean of Y in region R_j, and $I(x \in R_j)$ is an indicator function.

12.2.2 Building the Tree

The tree is built by recursively splitting the data:

1. At each step, split the data into two regions by choosing the predictor X_j and cutoff s that minimize the Residual Sum of Squares (RSS):

$$RSS = \sum_{i:x_i \in R_1(j,s)} (y_i - \bar{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \bar{y}_{R_2})^2.$$

2. Repeat the process within each resulting region until a stopping criterion is met (e.g., a minimum number of observations in a region).

12.2.3 Pruning the Tree

Overly complex trees may overfit the data. To improve generalization, trees are pruned:

1. Grow a large tree using recursive splitting.

2. Use cost-complexity pruning to balance model complexity and accuracy:

$$CART = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \bar{y}_{R_j})^2 + \alpha J,$$

   where α controls the tradeoff between tree size and fit.

3. Select α via cross-validation.

12.3 Classification Trees

12.3.1 Definition

Classification trees predict a qualitative response by assigning the majority class in each region:

$$\hat{Y} = Mode(Y_{R_j}),$$

where Y_{R_j} are the training observations in region R_j.

12.3.2 Splitting Criterion

The quality of a split is evaluated using metrics such as:

- **Gini Index:**

$$G = \sum_{k=1}^{K} p_k(1 - p_k),$$

  where p_k is the proportion of observations in class k.

- **Entropy:**

$$H = -\sum_{k=1}^{K} p_k \log(p_k),$$

  where p_k is the class proportion in a region.

- **Misclassification Error:**

$$E = 1 - \max_k(p_k),$$

  where p_k is the proportion of the majority class.

12.4 Advantages and Limitations of Decision Trees

- **Advantages:**
  - Easy to interpret and visualize.
  - Handles both numerical and categorical data.
  - Non-linear relationships are captured automatically.

- **Limitations:**
  - Prone to overfitting without pruning.
  - High variance—small changes in data can lead to very different trees.
  - Typically less accurate than ensemble methods (e.g., random forests, boosting).

12.5 Extensions of Decision Trees

12.5.1 Random Forests

Random forests improve predictive performance by averaging over multiple trees:

- Each tree is grown on a bootstrap sample of the training data.
- At each split, only a random subset of predictors is considered.

12.5.2 Boosting

Boosting builds an additive model by sequentially fitting trees to the residuals of the current model:

$$f(x) = \sum_{m=1}^{M} \lambda h_m(x),$$

where $h_m(x)$ is the m-th tree and λ controls the learning rate.

12.6 Applications of Decision Trees

Decision trees are widely used for:

- Predictive modeling in regression and classification tasks.
- Exploratory data analysis to identify important predictors.
- Non-linear and interaction effects in data.

13 Tree-Based Methods: Bagging and Boosting

Bagging and boosting are ensemble methods that combine multiple decision trees to improve prediction accuracy and reduce overfitting. These methods leverage the power of multiple models to make predictions more robust.

13.1 Bagging (Bootstrap Aggregating)

Bagging improves model accuracy by reducing variance through the aggregation of predictions from multiple models.

13.1.1 Procedure

1. Generate B bootstrap samples from the training data.
2. Train a decision tree on each bootstrap sample.
3. For regression, average the predictions:

$$\hat{f}_{bag}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^b(x),$$

   where $\hat{f}^b(x)$ is the prediction from the b-th tree.

4. For classification, use majority voting:

$$\hat{y}_{bag} = Mode\big(\{\hat{y}^1, \hat{y}^2, \ldots, \hat{y}^B\}\big).$$

13.1.2 Key Characteristics

- Trees are grown deeply (without pruning) to minimize bias.
- Variance is reduced by averaging over many uncorrelated trees.

13.1.3 Random Forests: An Extension of Bagging

Random forests add an extra layer of randomness to bagging by selecting a random subset of predictors at each split:

- This decorrelates the trees and further reduces variance.
- Typically, $\sqrt{p}$ predictors are considered at each split for classification and $p/3$ for regression.

13.2 Boosting

Boosting is a sequential ensemble method that builds an additive model by iteratively fitting trees to the residuals of the previous model.

13.2.1 Procedure

1. Initialize the model with a constant value, e.g., the mean of Y for regression:

$$f_0(x) = \arg\min_c \sum_{i=1}^{n} L(y_i, c),$$

   where L is the loss function.

2. For $m = 1, \ldots, M$:

(a) Compute the residuals:

$$r_i^{(m)} = y_i - f_{m-1}(x_i).$$

(b) Fit a decision tree $h_m(x)$ to the residuals.

(c) Update the model:

$$f_m(x) = f_{m-1}(x) + \lambda h_m(x),$$

where λ is the learning rate.

3. Output the final model:

$$f(x) = \sum_{m=1}^{M} \lambda h_m(x).$$

13.2.2 Key Characteristics

- Each tree attempts to correct the errors of the previous one.
- Trees are typically shallow (e.g., 1-5 splits) to prevent overfitting.
- The learning rate λ controls the contribution of each tree, with smaller λ leading to better generalization but requiring more trees.

13.3 Comparison: Bagging vs. Boosting

- **Bagging:**
  - Reduces variance by averaging predictions.
  - Trees are trained independently.
  - Performs well when predictors are uncorrelated.
- **Boosting:**
  - Reduces bias by iteratively refining the model.
  - Trees are trained sequentially.
  - Performs well when a strong relationship exists between predictors and response.

13.4 Tuning Parameters for Boosting

Key parameters to tune for boosting:

- **Number of Trees (M):** Controls the complexity of the model. Larger M reduces bias but may increase overfitting.
- **Learning Rate (λ):** Controls the contribution of each tree. Smaller λ requires more trees but reduces overfitting.
- **Tree Depth:** Shallow trees (low depth) reduce variance and prevent overfitting.

13.5 Advantages and Limitations

- **Advantages:**
  - Both bagging and boosting improve predictive performance.
  - Random forests and boosted trees handle high-dimensional data and interactions effectively.
- **Limitations:**
  - Bagging and random forests are less interpretable due to the ensemble structure.
  - Boosting can be prone to overfitting if the learning rate and number of trees are not properly tuned.
  - Both methods are computationally intensive for large datasets.

13.6 Applications of Bagging and Boosting

Bagging and boosting are widely used for:

- Classification tasks (e.g., random forests for image recognition).
- Regression tasks (e.g., predicting housing prices with boosted trees).
- Handling complex relationships in data with high-dimensional features.

14 Support Vector Machines (SVMs)

Support Vector Machines (SVMs) are supervised learning methods used for both classification and regression. They are particularly effective for high-dimensional datasets and problems with non-linear decision boundaries.

14.1 Maximal Margin Classifier

The Maximal Margin Classifier is the simplest SVM and is used for binary classification in linearly separable data.

14.1.1 Definition

Given a dataset of n observations $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ where $y_i \in \{-1, +1\}$, the goal is to find a hyperplane that separates the two classes and maximizes the margin, defined as the minimum distance from any observation to the hyperplane.

The hyperplane is represented as:

$$f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p,$$

where:

$$Margin = \frac{2}{\|\beta\|}.$$

14.1.2 Optimization Problem

The maximal margin classifier solves the following optimization problem:

$$\min_{\beta_0,\beta} \frac{1}{2}\|\beta\|^2$$

subject to:

$$y_i(\beta_0 + \beta^\top x_i) \geq 1 \quad for all i.$$

14.1.3 Limitations

- Only applicable when the data is linearly separable.
- Highly sensitive to outliers.

14.2 Support Vector Classifier (Soft-Margin Classifier)

The Support Vector Classifier extends the maximal margin classifier to handle non-linearly separable data by introducing a soft margin.

14.2.1 Optimization Problem

The optimization problem is modified to allow for misclassified observations:

$$\min_{\beta_0,\beta,\xi} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n} \xi_i,$$

subject to:

$$y_i(\beta_0 + \beta^\top x_i) \geq 1 - \xi_i \quad and \quad \xi_i \geq 0 \quad for all i,$$

where:

- ξ_i: Slack variables allowing margin violations.
- C: Tuning parameter controlling the tradeoff between margin width and misclassification.

14.2.2 Key Features

- Points close to or within the margin are called *support vectors*.
- The decision boundary is determined by the support vectors, not the entire dataset.

14.3 Support Vector Machines for Non-Linear Boundaries

SVMs use kernel functions to transform the feature space into a higher dimension, allowing for non-linear decision boundaries.

14.3.1 Kernel Functions

A kernel function $K(x, x')$ measures the similarity between two observations x and x'. Common kernels include:

- **Linear Kernel:** $K(x, x') = x^\top x'$.
- **Polynomial Kernel:** $K(x, x') = (1 + x^\top x')^d$, where d is the degree.
- **Radial Basis Function (RBF) Kernel:** $K(x, x') = \exp(-\gamma \|x - x'\|^2)$, where γ controls the kernel width.

14.3.2 Optimization Problem with Kernels

The kernelized SVM solves the same optimization problem as the support vector classifier, but the feature space is implicitly transformed using the kernel function.

14.4 Tuning Parameters in SVMs

The key parameters to tune in SVMs are:

- C: Controls the tradeoff between a wide margin and fewer margin violations.
- Kernel parameters (e.g., d for polynomial kernel, γ for RBF kernel).

Cross-validation is used to select optimal parameter values.

14.5 Advantages and Limitations of SVMs

- **Advantages:**
  - Effective in high-dimensional spaces.
  - Works well for both linear and non-linear decision boundaries.
  - Robust to overfitting, especially with proper tuning.
- **Limitations:**
  - Computationally intensive for large datasets.
  - Choice of kernel and tuning parameters significantly affects performance.
  - Less interpretable than simpler models like logistic regression.

14.6 Applications of SVMs

SVMs are widely used in:

- Text classification and sentiment analysis.
- Image recognition and object detection.
- Bioinformatics for protein classification and gene expression analysis.

15 Unsupervised Learning

Unsupervised learning involves analyzing data without labeled responses. The goal is to discover patterns or structures in the data, such as clusters or associations. Common techniques include clustering and dimension reduction.

15.1 Clustering Methods

Clustering involves partitioning observations into groups (clusters) based on their similarity.

15.1.1 K-Means Clustering

K-means clustering aims to partition the data into K clusters by minimizing the within-cluster variation.

Algorithm

1. Randomly initialize K cluster centroids.
2. Assign each observation to the nearest cluster centroid.
3. Update the cluster centroids by computing the mean of observations in each cluster.
4. Repeat steps 2 and 3 until the cluster assignments stabilize or a maximum number of iterations is reached.

Objective Function The algorithm minimizes the total within-cluster sum of squares (WCSS):

$$WCSS = \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - \mu_k\|^2,$$

where C_k is the set of observations in cluster k, and μ_k is the centroid of cluster k.

Limitations

- Sensitive to the choice of K.
- May converge to a local minimum; results depend on initialization.

15.1.2 Hierarchical Clustering

Hierarchical clustering creates a tree-based representation (dendrogram) of the data, which does not require specifying the number of clusters in advance.

Types

- **Agglomerative (Bottom-Up):** Start with each observation as its own cluster and iteratively merge the closest clusters.
- **Divisive (Top-Down):** Start with all observations in one cluster and iteratively split the clusters.

Distance Metrics The similarity between clusters is measured using:

- **Complete Linkage:** Maximum distance between observations in two clusters.
- **Single Linkage:** Minimum distance between observations in two clusters.
- **Average Linkage:** Average distance between observations in two clusters.
- **Centroid Linkage:** Distance between centroids of two clusters.

Advantages and Limitations

- **Advantages:** Visual representation via dendrograms; no need to specify K in advance.
- **Limitations:** Computationally expensive for large datasets.

15.2 Principal Component Analysis (PCA)

PCA is a dimension reduction technique that identifies directions (principal components) capturing the maximum variance in the data.

15.2.1 Definition

Given a dataset X with n observations and p variables, PCA transforms the data into K principal components:

$$Z_k = \sum_{j=1}^{p} \phi_{jk} X_j,$$

where ϕ_{jk} are the loadings of the k-th principal component.

15.2.2 Objective

PCA maximizes the variance captured by each principal component:

$$Var(Z_k) = \lambda_k,$$

where λ_k is the eigenvalue of the k-th principal component. The total variance is the sum of all eigenvalues:

$$TotalVariance = \sum_{k=1}^{p} \lambda_k.$$

15.2.3 Applications

- Visualizing high-dimensional data.
- Reducing dimensionality for clustering or regression.
- Removing noise by discarding components with low variance.

15.2.4 Limitations

- Assumes linear relationships between variables.
- Sensitive to scaling of variables.

15.3 Applications of Unsupervised Learning

Unsupervised learning is widely used in:

- Market segmentation (e.g., clustering customers based on purchasing behavior).
- Image compression (e.g., PCA for reducing image dimensions).
- Gene expression analysis (e.g., clustering genes or samples with similar profiles).

15.4 Comparison of Clustering and Dimension Reduction

- **Clustering:** Groups observations into meaningful clusters; focuses on relationships between data points.
- **Dimension Reduction:** Reduces the number of variables while preserving variance; focuses on relationships between variables.