EE353: Midsem notes (Main)

Sravan K Suresh

September 19, 2024

1. IID Assumption

The **IID (Independent and Identically Distributed)** assumption means:

- **Independent**: Each data point is drawn independently.
- **Identically Distributed**: Each data point comes from the same probability distribution.

This assumption underlies many statistical models and tests.

2. Data Log Likelihood

The **log likelihood** measures the fit of a statistical model to the data:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} \log P(x_i|\theta)$$

- x_i: individual data point.
- θ: model parameters.
- Maximizing log likelihood helps estimate the best parameters for the model.

3. Comparing Distributions

Comparing distributions is essential to understand how different datasets or variables behave. Common techniques include:

- Visual comparison (e.g., histograms, box plots).
- Statistical tests (e.g., t-tests, rank-sum test).

4. Confidence Interval Around a Mean

A confidence interval gives a range within which the true population mean lies with a certain probability:

$$CI = \bar{x} \pm z \cdot \frac{s}{\sqrt{n}}$$

- $\bar{x}$: sample mean.
- z: z corresponding to the confidence level.
- s: sample standard deviation.
- n: sample size.

5. Independent t-test (Ignoring Degrees of Freedom)

The **independent t-test** compares the means of two independent samples:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

- $\bar{x}_1, \bar{x}_2$: sample means.
- s_1, s_2: sample standard deviations.
- n_1, n_2: sample sizes.

6. Paired t-test (Ignoring Degrees of Freedom)

The **paired t-test** is used to compare two related samples:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}}$$

- $\bar{d}$: mean of the differences between paired observations.
- s_d: standard deviation of the differences.
- n: number of pairs.

7. Wilcoxon Rank-Sum Test

A non-parametric test that compares two independent samples, it assesses whether their populations have the same distribution. It's an alternative to the independent t-test:

$$W = \text{sum of ranks of one sample}$$

8. Wilcoxon Signed-Rank Test

A non-parametric test for comparing two related samples, this test is an alternative to the paired t-test:

$$W = \text{sum of ranks of the differences between paired observations}$$

9. Pearson's Correlation Coefficient

Pearson's correlation measures the linear relationship between two variables:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

10. Spearman's Rank Correlation

Spearman's correlation assesses the monotonic relationship between two ranked variables:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

- d_i: difference between the ranks of corresponding variables.
- Does not assume normality and is useful for ordinal data.

11. Basic Linear Regression Model

The basic linear regression model assumes the relationship between input variables X and target variable y is linear:

$$y = X\beta + \epsilon$$

- $X \in \mathbb{R}^{n \times p}$: matrix of input features. - $\beta \in \mathbb{R}^{p}$: vector of coefficients. - ϵ: noise term (error).

12. Noise Assumption

The error term ϵ is assumed to be independent and normally distributed:

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

- This assumption allows us to derive properties of the model and use maximum likelihood estimation.

13. Derivation of MSE as Loss Function

From the probabilistic assumption, the likelihood of observing the data y given X and β is:

$$P(y|X, \beta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - X_i\beta)^2}{2\sigma^2}\right)$$

Maximizing the log-likelihood leads to the minimization of the Mean Squared Error (MSE) as the loss function:

$$\text{MSE}(\beta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - X_i\beta)^2$$

14. Pseudoinverse and Maximum Likelihood Solution

The optimal solution to linear regression can be found via the pseudoinverse:

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

This is also the **maximum likelihood estimator** (MLE) under the assumption of normally distributed errors.

15. Extension to MAE (Laplace Error Distribution)

When the error distribution follows a Laplace distribution:

$$P(\epsilon) = \frac{1}{2b} \exp\left(-\frac{|y - X\beta|}{b}\right)$$

Minimizing the negative log-likelihood results in the **Mean Absolute Error (MAE)**:

$$\text{MAE}(\beta) = \frac{1}{n}\sum_{i=1}^{n}|y_i - X_i\beta|$$

MAE is robust to outliers compared to MSE.

16. Bias-Variance-Noise Decomposition

The expected loss of a model can be decomposed into three components:

$$\text{Expected Loss} = \text{Bias}^2 + \text{Variance} + \text{Noise}$$

17. L2 (Ridge) and L1 (Lasso) Penalties

L2 Regularization (Ridge)

L2 regularization adds a penalty on the square of the coefficients:

$$\text{L2 Penalty} = \lambda \|\beta\|_2^2 = \lambda \sum_{j=1}^{p} \beta_j^2$$

This prevents large coefficient values and helps in handling multicollinearity.

L1 Regularization (Lasso)

L1 regularization adds a penalty on the absolute values of the coefficients:

$$\text{L1 Penalty} = \lambda \|\beta\|_1 = \lambda \sum_{j=1}^{p} |\beta_j|$$

L1 regularization can lead to sparse solutions by driving some coefficients to zero, effectively performing variable selection.

18. Geometry of L2 and L1 Regularization

L2 Regularization Geometry

L2 regularization constrains the solution within a **circular** region (due to the Euclidean norm). This geometry does not lead to sparse solutions but shrinks coefficients uniformly.

L1 Regularization Geometry

L1 regularization constrains the solution within a **diamond-shaped** region. The sharp corners of the diamond lead to variable elimination (where coefficients become exactly zero).

Gradient Descent for L2 vs L1

19. Gradient Descent for L2 (Ridge)

The gradient of the L2 regularized objective function is:

$$\nabla = -\frac{2}{n} X^T (y - X\beta) + 2\lambda\beta$$

- The penalty term is smooth, making it easier to optimize using gradient descent.

20. Gradient Descent for L1 (Lasso)

The gradient of the L1 regularized objective function is:

$$\nabla = -\frac{2}{n} X^T (y - X\beta) + \lambda \text{sign}(\beta)$$

- The L1 penalty introduces non-differentiability at zero, making it harder to optimize with gradient descent (sub-gradient methods are often used).

1. Basic Decision Criteria

In linear classification, the decision boundary is a linear function of the input. The classifier outputs class labels based on the sign of a linear combination of input features:

$$f(x) = \text{sign}(w^T x + b)$$

- $w \in \mathbb{R}^d$ is the weight vector. - b is the bias term. - x is the feature vector.

The decision rule is:

$$\hat{y} = \begin{cases} 1 & \text{if } w^T x + b \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

2. Bayesian Classifier for Gaussian Class Conditionals

When class-conditional distributions are Gaussian with the same covariance matrix, the optimal linear classifier is derived as follows.

Gaussian Assumptions

Let the class-conditional distributions be:

$$P(x|y=0) = \mathcal{N}(\mu_0, \Sigma), \quad P(x|y=1) = \mathcal{N}(\mu_1, \Sigma)$$

The posterior probability for class y given x is:

$$P(y=1|x) = \frac{P(x|y=1)P(y=1)}{P(x|y=0)P(y=0) + P(x|y=1)P(y=1)}$$

Linear Decision Boundary

Taking the log-odds and simplifying under the assumption of equal covariance matrices, we obtain the linear decision boundary:

$$w = \Sigma^{-1}(\mu_1 - \mu_0), \quad b = -\frac{1}{2}(\mu_1^T \Sigma^{-1} \mu_1 - \mu_0^T \Sigma^{-1} \mu_0) + \log \frac{P(y=1)}{P(y=0)}$$

This gives a linear classifier of the form:

$$f(x) = \text{sign}(w^T x + b)$$

3. Bayesian Classifier for Non-Gaussian Class Conditionals

When class conditionals are non-Gaussian, we extend the Bayesian framework by:

- Estimating the posterior using non-parametric methods (e.g., kernel density estimation).
- Modeling class-conditional distributions using non-Gaussian distributions such as mixtures of Gaussians or other distributions depending on the data.

The general rule remains:

$$\hat{y} = \text{argmax}_y \; P(y|x)$$

4. Logistic Regression

Logistic regression models the probability of class membership using the logistic (sigmoid) function:

$$P(y = 1|x) = \sigma(w^T x + b) = \frac{1}{1 + \exp(-(w^T x + b))}$$

The decision boundary is:

$$\hat{y} = \begin{cases} 1 & \text{if } P(y = 1|x) \geq 0.5, \\ 0 & \text{otherwise.} \end{cases}$$

Logistic Regression Loss Function

Logistic regression minimizes the negative log-likelihood:

$$L(w, b) = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log P(y_i|x_i) + (1 - y_i) \log(1 - P(y_i|x_i))]$$

This is equivalent to minimizing the cross-entropy between the predicted probabilities and true labels.

5. Gradient Descent for Logistic Regression

The gradient of the logistic loss function with respect to the weights w and bias b is:

$$\nabla_w L = -\frac{1}{n} \sum_{i=1}^{n} (y_i - \sigma(w^T x_i + b)) x_i$$

$$\nabla_b L = -\frac{1}{n} \sum_{i=1}^{n} (y_i - \sigma(w^T x_i + b))$$

Using gradient descent, we iteratively update the weights and bias:

$$w^{(t+1)} = w^{(t)} - \eta \nabla_w L$$

$$b^{(t+1)} = b^{(t)} - \eta \nabla_b L$$

where η is the learning rate.

6. L2 Regularization in Logistic Regression

To prevent overfitting, we can add an L2 penalty to the logistic regression objective function:

$$L_{\text{reg}}(w, b) = L(w, b) + \frac{\lambda}{2} \|w\|_2^2$$

The regularized loss function becomes:

$$L_{\text{reg}}(w, b) = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log P(y_i|x_i) + (1 - y_i) \log(1 - P(y_i|x_i))] + \frac{\lambda}{2} \|w\|_2^2$$

The gradient for L2 regularized logistic regression is:

$$\nabla_w L_{\text{reg}} = \nabla_w L + \lambda w$$