Predictive Solutions

Problem 1

(a) (1 point)

In logistic regression, what distinguishes the probit link function from the logit link function?

The key distinction lies in the assumed distribution of the error term:

- **Logit link function:** Assumes that the error term follows a logistic distribution. The probability model is:

$$P(y = 1 \mid x) = \frac{\exp(x^T \beta)}{1 + \exp(x^T \beta)}.$$

- **Probit link function:** Assumes that the error term follows a standard normal distribution. The probability model is:

$$P(y = 1 \mid x) = \Phi(x^T \beta),$$

where $\Phi(\cdot)$ is the cumulative distribution function (CDF) of the standard normal distribution.

(b) (2 points)

Briefly explain the relationship between cross-entropy and the negative log-likelihood function.

The **cross-entropy loss** in machine learning is equivalent to the **negative log-likelihood** (NLL) in statistical modeling for classification tasks. Specifically:

- For a binary classification task with $P(y = 1 \mid x) = p(x)$, the NLL is given by:

$$\text{NLL} = -\sum_{i=1}^{n} \left[y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i)) \right].$$

- The cross-entropy loss has the same mathematical form and measures how well the predicted probabilities match the true labels.

(c) (2 points)

Why will the performance of linear discriminant analysis and logistic regression be equivalent for the same feature space?

Both methods assume linear decision boundaries:

- **Logistic regression:** Models $P(y = 1 \mid x)$ directly using a logit link.
- **Linear discriminant analysis (LDA):** Assumes that features are normally distributed within each class and derives linear boundaries under this assumption.

When the assumptions of LDA hold, both methods yield equivalent decision boundaries.

(d) (2 points)

Define distributed multinomial logistic regression for a k-class classification problem.

Distributed multinomial logistic regression splits the computation across multiple nodes for scalability. The general model for k-class classification is:

$$P(y = j \mid x) = \frac{\exp(x^T \beta_j)}{\sum_{l=1}^{k} \exp(x^T \beta_l)}, \quad j = 1, 2, \ldots, k.$$

- In the distributed setup, data is divided across nodes, and each node computes partial likelihoods.
- The results are aggregated iteratively to update parameters using algorithms like distributed gradient descent.

(e) (3 points)

For categorical time series data $D = \{y_t : t = 1, 2, \ldots, T\}$, where $y_t = 0$ with probability p_t and $y_t = 1$ with probability $1 - p_t$, develop an auto-regressive time series model of order p within a logistic regression framework.

The auto-regressive model incorporates past values of y_t:

$$\text{logit}(p_t) = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \ldots + \beta_p y_{t-p},$$

where $p_t = P(y_t = 1 \mid y_{t-1}, y_{t-2}, \ldots, y_{t-p})$.

This model captures temporal dependencies in the categorical time series.

Problem 2

The linear discriminant rule for class k is:

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k.$$

The best decision rule is:

$$G(x) = \arg\max_k \delta_k(x).$$

Given:

$$\bar{X}_1 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad \bar{X}_2 = \begin{bmatrix} 5 \\ 7 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, \quad n_1 = 20, \quad n_2 = 40.$$

Logarithms: $\log(2) = 0.693$, $\log(3) = 1.099$.

(i) (5 points) Compute the linear discriminant function and classify the point $x = [1, 4]$.

We compute the discriminant functions for both classes:

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k.$$

1. **Calculate Σ^{-1}:**

$$\Sigma = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, \quad \det(\Sigma) = (1)(2) - (1)(1) = 1.$$

$$\Sigma^{-1} = \frac{1}{\det(\Sigma)} \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}.$$

2. **Compute π_1 and π_2:**

$$\pi_1 = \frac{n_1}{n_1 + n_2} = \frac{20}{60} = \frac{1}{3}, \quad \pi_2 = \frac{n_2}{n_1 + n_2} = \frac{40}{60} = \frac{2}{3}.$$

$$\log(\pi_1) = \log\left(\frac{1}{3}\right) = -1.099, \quad \log(\pi_2) = \log\left(\frac{2}{3}\right) = -0.405.$$

3. **Compute $\delta_1(x)$:**

$$\mu_1 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad \mu_1^T \Sigma^{-1} \mu_1 = \begin{bmatrix} 2 & 3 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

$$\mu_1^T \Sigma^{-1} \mu_1 = (2)(2) + (3)(-1) + (2)(-1) + (3)(1) = 4 - 3 - 2 + 3 = 2.$$

$$x^T \Sigma^{-1} \mu_1 = \begin{bmatrix} 1 & 4 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

$$x^T \Sigma^{-1} \mu_1 = (1)(2) + (4)(-1) + (1)(-1) + (4)(1) = 2 - 4 - 1 + 4 = 1.$$

$$\delta_1(x) = 1 - \frac{1}{2}(2) + (-1.099) = 1 - 1 - 1.099 = -1.099.$$

4. **Compute $\delta_2(x)$:** Similarly, compute:

$$\mu_2^T \Sigma^{-1} \mu_2, \quad x^T \Sigma^{-1} \mu_2, \quad \delta_2(x).$$

Classification: Assign $x = [1, 4]$ to the class with the higher $\delta_k(x)$.

(ii) (5 points) Under which theoretical assumptions do you expect your method to be reliable?

1. **Assumption of Normality:** The features in each class are normally distributed.
2. **Equal Covariance Matrices:** All classes share the same covariance matrix (Σ).
3. **Independence:** The observations are independent.
4. **Large Sample Size:** Reliable parameter estimates require sufficiently large sample sizes.

Problem 3

(a) (3 points) When is the Gaussian process prior regression model useful?

Gaussian process (GP) regression is particularly useful when:

1. The underlying function is unknown and potentially non-linear.
2. The data is limited, as GP models provide uncertainty quantification.
3. Interpretability and flexibility are required, as the covariance function encodes prior knowledge.

(b) (4 points) Modify the Gaussian process prior regression for a binary classification problem.

For binary classification, GP regression is adapted by using a probit or logit link function:

1. **Likelihood:** For binary $y \in \{0, 1\}$, the likelihood becomes:

   $$P(y = 1 \mid f(x)) = \Phi(f(x)),$$

   where $f(x)$ is the latent function and $\Phi(\cdot)$ is the standard normal CDF.

2. **Prior:** The latent function $f(x)$ is modeled as a Gaussian process:

   $$f(x) \sim \mathcal{GP}(0, k(x, x')),$$

   where $k(x, x')$ is the covariance function.

(c) (3 points) Why does the Gaussian process prior model fail for big data?

GP models fail for big data because:

1. **Computational Complexity:** Matrix inversion in GPs scales as $O(n^3)$, making it infeasible for large datasets.
2. **Memory Requirements:** Storage of the covariance matrix requires $O(n^2)$ memory.

Problem 4

(a) (3 points) Briefly explain the relationship between the basis expansion technique in functional regression and feature engineering in machine learning.

- **Basis Expansion in Functional Regression:** Represents a complex function $f(x)$ as a linear combination of simpler basis functions:

$$f(x) = \sum_{j=1}^{K} \beta_j \phi_j(x),$$

  where $\phi_j(x)$ are basis functions (e.g., polynomials, splines, Fourier basis). This transforms the problem into linear regression in the transformed feature space.

- **Feature Engineering in Machine Learning:** Creates new features by transforming raw inputs, often using domain knowledge or mathematical techniques. For example:
  - Adding polynomial terms to capture non-linearity.
  - Using Fourier transforms for periodic data.

Relationship: Both methods transform input features to enable models to better capture underlying relationships, improving model accuracy and flexibility.

(b) (2 points) Briefly explain the Kosambi–Karhunen–Loève theorem in the context of Gaussian process prior.

The Kosambi–Karhunen–Loève (KKL) theorem states that any stochastic process $f(x)$ with finite variance can be expressed as an infinite sum of orthogonal functions:

$$f(x) = \sum_{j=1}^{\infty} \beta_j \phi_j(x),$$

where:

- $\phi_j(x)$ are orthogonal basis functions.
- β_j are coefficients determined by the process.

In Gaussian Process Priors:

1. Gaussian processes leverage the KKL decomposition, using a covariance kernel to derive orthogonal basis functions.
2. This ensures that the induced stochastic process $f(x)$ is smooth and satisfies the given covariance structure.

(c) (2 points) Explain the relationship between the distance between points and the squared-exponential covariance function of a Gaussian process model.

The squared-exponential covariance function is:

$$k(x, x') = \sigma^2 \exp\left(-\frac{\|x - x'\|^2}{2l^2}\right),$$

where $\|x - x'\|$ is the distance between x and x'.

- When $\|x - x'\|$ is small (points are close), $k(x, x') \approx \sigma^2$, meaning the two points have similar values.
- When $\|x - x'\|$ is large, $k(x, x') \to 0$, implying no correlation.

Thus, the covariance depends inversely on the distance, controlled by the length scale l.

(d) (3 points) Gauss-Markov Linear Models are special cases of Gaussian Process Regression Models. Prove or Disprove the above statement.

Proof:

1. **Gaussian Process Regression (GPR):**
   - A GPR assumes $f(x) \sim \mathcal{GP}(0, k(x, x'))$, where the kernel $k(x, x')$ determines the covariance structure.
   - Predictions are made by conditioning on observed data.
2. **Gauss-Markov Linear Model:**
   - Assumes $y = X\beta + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$.
   - This can be viewed as a special case of GPR with:
     - Kernel $k(x, x') = x^T x'$ (linear kernel).
     - Mean function $f(x) = X\beta$.

Thus, Gauss-Markov Linear Models are special cases of GPR with a linear kernel.

Problem 5

(a) (3 points) What is tree-structured regression?

Tree-structured regression partitions the feature space into regions using decision trees:

1. The feature space is recursively split based on rules (e.g., $x_j \leq c$).
2. Each terminal node represents a region where predictions are made using the average of responses in that region.

Advantages:

- Non-parametric and flexible.
- Captures interactions between features automatically.

(b) (3 points) What is the appropriate objective function for tree-structured regression? How to fit a tree-structured regression?

Objective Function: Minimize the sum of squared residuals (SSR) within each region:

$$\text{SSR} = \sum_{j=1}^{M} \sum_{i \in R_j} (y_i - \bar{y}_{R_j})^2,$$

where M is the number of regions, R_j is the j-th region, and $\bar{y}_{R_j}$ is the mean response in R_j.

Fitting Process:

1. Start with the entire feature space.
2. At each step, split the space to minimize the SSR.
3. Stop splitting based on criteria (e.g., minimum node size or maximum tree depth).

(c) (4 points) What is intrinsic feature selection? How does tree-structured regression achieve intrinsic feature selection?

Intrinsic Feature Selection: A model automatically selects the most important features during training, without explicit regularization.

In Tree-Structured Regression:

- Splits are made based on features that most reduce the SSR.
- Irrelevant features are ignored since they don't contribute to meaningful splits.

This automatic selection of relevant features makes decision trees inherently robust to irrelevant variables.

Problem 6

(a) (2 points) How can a non-monotonic relationship be captured in logistic regression?

Non-monotonic relationships can be captured by:

1. **Feature Engineering:** Adding higher-order polynomial terms (e.g., x^2, x^3) to the model:
$$\text{logit}(p) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3.$$
2. **Non-linear Transformations:** Applying transformations (e.g., $\sin(x)$, splines) to the features.

(b) (2 points) Briefly explain the relationship between logistic regression and neural networks.

1. **Similarity:** Logistic regression is equivalent to a single-layer perceptron with a sigmoid activation function.
2. **Extension:** Neural networks generalize logistic regression by adding hidden layers and non-linear activation functions, allowing modeling of complex relationships.

(c) (2 points) How can you estimate λ in Ridge regression?

λ can be estimated using:

1. **Cross-Validation:** Split the data, fit models for different λ values, and select the one minimizing the validation error.
2. **Information Criteria:** Use AIC or BIC to choose λ that balances fit and complexity.

(d) (4 points) Why does Fourier Basis expansion not suffer from the multicollinearity problem?

Fourier basis functions (e.g., $\sin(kx)$, $\cos(kx)$) are orthogonal. This means:

$$\int \sin(kx)\sin(k'x)dx = 0, \quad \text{for } k \neq k',$$

ensuring no linear dependence between terms. Orthogonality prevents multicollinearity.

Problem 7

The equations of the regression lines are:

$$Y - \bar{Y} = r\frac{s_y}{s_x}(X - \bar{X}),$$

$$X - \bar{X} = r\frac{s_x}{s_y}(Y - \bar{Y}),$$

where:

- $\bar{X}, \bar{Y}$: Sample means of X, Y.
- s_x, s_y: Sample standard deviations of X, Y.
- r: Sample correlation coefficient.

(a) (4 points) If θ is the acute angle between the two regression lines, then obtain θ in terms of r, s_x, s_y.

1. **Slopes of the lines:**
$$m_1 = r\frac{s_y}{s_x}, \quad m_2 = \frac{1}{m_1} = r\frac{s_x}{s_y}.$$

2. **Angle θ:** The angle between the lines is given by:
$$\tan\theta = \left|\frac{m_1 - m_2}{1 + m_1 m_2}\right|.$$

3. **Substitute m_1 and m_2:**
$$m_1 m_2 = \left(r\frac{s_y}{s_x}\right)\left(r\frac{s_x}{s_y}\right) = r^2.$$
$$\tan\theta = \left|\frac{r\frac{s_y}{s_x} - r\frac{s_x}{s_y}}{1 + r^2}\right| = \left|\frac{r\left(\frac{s_y}{s_x} - \frac{s_x}{s_y}\right)}{1 + r^2}\right|.$$

4. **Simplify:**
$$\tan\theta = \left|\frac{r\frac{s_y^2 - s_x^2}{s_x s_y}}{1 + r^2}\right|.$$

5. **Final Result:**
$$\theta = \arctan\left(\left|\frac{r(s_y^2 - s_x^2)}{s_x s_y(1 + r^2)}\right|\right).$$

(b) (3 points) If $r = 0$, then find θ.

When $r = 0$, the slopes $m_1 = 0$ and $m_2 = 0$. Thus, the regression lines are perpendicular, and:
$$\theta = 90^\circ \quad \text{or} \quad \theta = \frac{\pi}{2}.$$

(c) (3 points) If $r = 1$, then find θ.

When $r = 1$, the slopes $m_1 = m_2$. The regression lines coincide, and the angle between them is:
$$\theta = 0^\circ.$$

Problem 8

A variable star's brightness fluctuates over time. We are tasked with proposing regression models and a strategy for comparison.

(a) (4 points) Propose an appropriate regression model for predicting the brightness of a star.

Given the time-series nature of the data, a suitable model is:

Fourier Regression:

$$y(t) = \beta_0 + \sum_{k=1}^{K} \left[\beta_k \sin\left(\frac{2\pi k t}{P}\right) + \gamma_k \cos\left(\frac{2\pi k t}{P}\right) \right] + \epsilon,$$

where:

- P: Fundamental period of brightness variation.
- β_k, γ_k: Fourier coefficients.
- ϵ: Noise term.

This model captures periodic fluctuations observed in variable star brightness.

(b) (4 points) Propose an alternative regression model for consideration.

An alternative model is:

Gaussian Process Regression:

$$y(t) \sim \mathcal{GP}(\mu(t), k(t, t')),$$

where:

- $\mu(t)$: Mean function (e.g., a smooth trend over time).
- $k(t, t')$: Covariance function (e.g., squared-exponential kernel) to capture temporal dependencies.

This model is flexible and accommodates non-linear variations.

(c) (2 points) What would be your strategy for identifying the best and second-best models?

1. **Fit Both Models:** Train both the Fourier and Gaussian Process models using the data.
2. **Model Comparison Criteria:** Evaluate the models using:
   - **Akaike Information Criterion (AIC):** Penalizes model complexity while rewarding goodness of fit.
   - **Cross-Validation (CV):** Assess predictive performance on unseen data.
3. **Rank Models:**
   - Select the model with the lowest AIC or best CV score as the best model.
   - Choose the second-best based on the next lowest score.