Written Assignment 2

Yushan Liu   Student ID: 2024214103

November 16, 2024

Problem 2.1: SVM and Logitic Regression

(a) Definition of $E_\infty(z)$ and the regularization parameter λ

The function $E_\infty(z)$ is defined as:

$$E_\infty(z) = \begin{cases} 0 & \text{if } z \geq 1 \\ 1 - z & \text{if } 0 < z < 1 \\ \infty & \text{if } z \leq 0 \end{cases}$$

Here, z is the product $y^{(i)}(\omega^T x^{(i)} + b)$. This function indicates that:

- If the predicted output is correctly classified with a margin of at least 1(i.e., $z \geq 1$), the loss is 0.
- If the predicted output is within the margin(i.e., $0 < z < 1$), it incurs a linear penalty proportational to how far z is from to 1.
- If the predicted output is incorrrect(i.e., $z \leq 0$), the penalty is infinite, representing a hard constraint violation.

The regularization parameter λ controls the trade-off between maximizing the margin and minimizing the classification error. The constraint for λ is typically:

$$\lambda \geq 0$$

(b) Definition of $E_{LR}(z)$

Given a target variable $y \in \{-1, 1\}$, we know that the logistic regression model defines the probability of $y = 1$ given the feature vector x as:

$$p(y = 1|x) = \sigma(w^\top x + b)$$

where $\sigma(z)$ is the sigmoid function. Consequently the probability of $y = -1$ is:

$$p(y = -1|x) = 1 - p(y = 1|x) = 1 - \sigma(w^\top x + b) = \sigma(-(w^\top x + b))$$

The likelihood of observing the data for m samples $(x^{(i)}, y^{(i)})$ is given by:

$$L(w, b) = \prod_{i=1}^{m} p(y^{(i)}|x^{(i)})$$

For logistic regression, this can be written as:

$$L(w, b) = \prod_{i=1}^{m} \sigma(w^\top x^{(i)} + b)^{\frac{1+y^{(i)}}{2}} \cdot \sigma(-(w^\top x^{(i)} + b))^{\frac{1-y^{(i)}}{2}}$$

Thus the negative log-likelihood is:

$$-\log L(w, b) = -\sum_{i=1}^{m} \left(\frac{1 + y^{(i)}}{2} \log \sigma(w^\top x^{(i)} + b) + \frac{1 - y^{(i)}}{2} \log \sigma(-(w^\top x^{(i)} + b)) \right)$$

To incorporate the regularization term, we add $\lambda||w||^2$ to the negative log-likelihood, resulting in the following expression:

$$\sum_{i=1}^{m} E_{LR}(y^{(i)}(w^\top x^{(i)} + b)) + \lambda||w||^2$$

The function $E_{LR}(z)$ captures the negative log-likelihood for a single instance and is defined as follows:

$$E_{LR}(z) = \log(1 + e^{-z}) \quad \text{for } z = y^{(i)}(w^\top x^{(i)} + b)$$

In this case:

- If $y^{(i)} = 1$, $E_{LR}(z) = \log(1 + e^{-(w^\top x^{(i)}+b)})$.
- If $y^{(i)} = -1$, $E_{LR}(z) = \log(1 + e^{(w^\top x^{(i)}+b)})$.

(c) Definition of $E_{SV}(z)$ and the regularization parameter λ

The modified SVM optimization problem can be expressed in the desired form, we start by analyzing the introduction of slack variables $\xi^{(i)}$ in the constraints. The original constraints for hard-margin SVM are replaced by:

$$y^{(i)}(w^\top x^{(i)} + b) \geq 1 - \xi^{(i)}, \quad \text{for } i = 1, \dots, m,$$

where $\xi^{(i)} \geq 0$ accounts for misclassifications.The new objective function we want to minimize becomes:

$$C \sum_{i=1}^{m} \xi^{(i)} + \frac{1}{2}||w||^2.$$

To establish a relationship between $y^{(i)}(w^\top x^{(i)} + b)$ and $\xi^{(i)}$, we can write:

$$\xi^{(i)} = \max(0, 1 - y^{(i)}(w^\top x^{(i)} + b)).$$

This formulation shows that:

- If the data point is correctly classified with a margin of at least 1 (i.e., $y^{(i)}(w^\top x^{(i)} + b) \geq 1$), then $\xi^{(i)} = 0$.
- If the data point is on the wrong side of the margin (i.e., $y^{(i)}(w^\top x^{(i)} + b) < 1$), then $\xi^{(i)}$ quantifies the extent of misclassification.

The Lagrangian for the soft-margin SVM can be expressed as:

$$\mathcal{L}(w, b, \xi, \alpha) = \frac{1}{2}||w||^2 + C \sum_{i=1}^{m} \xi^{(i)} - \sum_{i=1}^{m} \alpha^{(i)} \left(y^{(i)}(w^\top x^{(i)} + b) - 1 + \xi^{(i)} \right),$$

where $\alpha^{(i)}$ are the Lagrange multipliers. Given the relationship established above, we can express the sum of slack variables as:

$$\sum_{i=1}^{m} \xi^{(i)} = \sum_{i=1}^{m} E_{SV}(y^{(i)}(w^\top x^{(i)} + b)),$$

where $E_{SV}(z)$ is defined as follows:

$$E_{SV}(z) = \begin{cases} 0 & \text{if } z \geq 1 \\ 1 - z & \text{if } 0 < z < 1 \\ -\frac{1}{2}z^2 + \frac{1}{2} & \text{if } z < 0 \end{cases}$$

This function captures the penalties incurred based on how far $y^{(i)}(w^\top x^{(i)}+b)$ deviates from the decision boundary.

The regularization parameter in this context is related to the trade-off between the classification error and the model complexity, and can be expressed as:

$$\lambda = C.$$

(d) Conclusion and Discussion

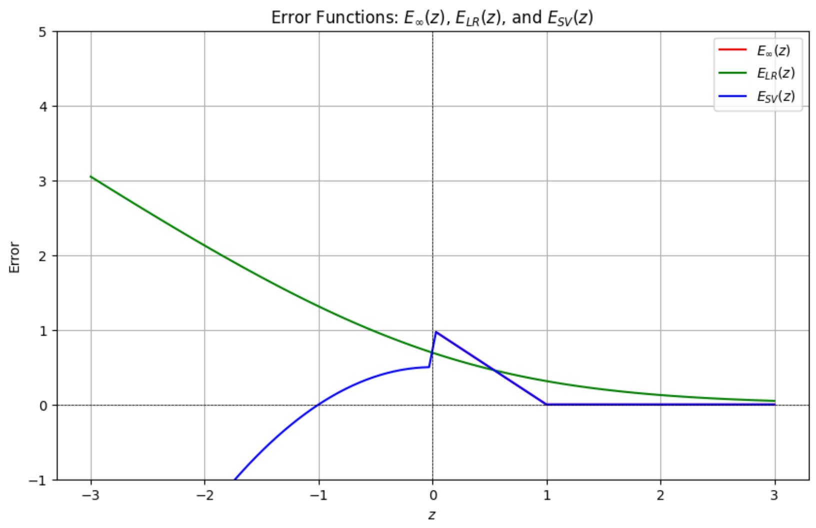
We can use Python libraires like `Numpy` and `Matplotlib` to plot the three error functions above, which inplementation code is shown below:

```
def E_inf(z):
    """Error function E∞(·)"""
    return np.where(z >= 1, 0, np.where(z > 0, 1 - z, np.inf))

def E_LR(z):
    """Error function ELR(·)"""
    return np.log(1 + np.exp(-z))

def E_SV(z):
    """Error function ESV(·)"""
    return np.where(z >= 1, 0, np.where(z > 0, 1 - z, -0.5 * z**2 + 0.5))
```

The three error functions are plotted in the figure below:



We can see that:

- $E_\infty(z)$: It exhibits a hard threshold; any misclassification incurs an infinite penalty. This is ideal for a clear margin but may not be suitable for noisy data.
- $E_{LR}(z)$: It is smooth and continuous, which allows for a softer transition between correct and incorrect classifications. This smoothness can help the optimization process, leading to better convergence properties.
- $E_{SV}(z)$: It balances balances misclassifications and model complexity. The quadratic penalty for $z < 0$ help avoid extreme penalities while still enforcing the margin constraints.

If the error functions are replaced with other functions, several outcomes may arise:

- **Loss of Robustness**: Using an overly aggressive loss function (like a hard-margin SVM) may lead to poor generalization on noisy or overlapping datasets, resulting in overfitting.
- **Convergence Issues**: Non-smooth functions could lead to difficulties in optimization, causing convergence problems or slow training.
- **Behavior in Overlapping Data**: For functions that do not adequately penalize misclassifications, such as squared error loss in classification tasks, the model may not handle overlap well, potentially leading to high error rates.
- **Interpretability and Application**: Different functions might have different interpretability and applications. For example, soft-margin SVM and logistic regression can be interpreted probabilistically, whereas a hard-margin approach may not.

Problem 2.2: Naive Bayes Parameter Learning

We are given a dataset $\{(x^{(i)}, y^{(i)}), i = 1, 2, \ldots, m\}$ where each $x^{(i)}$ is an n-dimensional vector, with each element $x_j^{(i)} \in \{0, 1\}$, and the labels $y^{(i)} \in \{0, 1\}$.We have known the following model for Naive Bayes classification:

$y^{(i)}$ follows a Bernoulli distribution:

$$y^{(i)} \sim \text{Bernoulli}(\phi_y)$$

Given $y^{(i)} = b$, the features $x_j^{(i)}$ are independent and follow Bernoulli distributions:

$$x_j^{(i)} | y^{(i)} = b \sim \text{Bernoulli}(\phi_{j|y=b}), \quad b = 0, 1$$

The joint probability distribution of a single data point $(x^{(i)}, y^{(i)})$ is:

$$p(x^{(i)}, y^{(i)} | \phi_y, \phi_{j|y=b}) = p(y^{(i)} | \phi_y) \cdot \prod_{j=1}^{n} p(x_j^{(i)} | y^{(i)}, \phi_{j|y=b})$$

Specifically:

- $p(y^{(i)} | \phi_y) = \phi_y^{y^{(i)}} (1 - \phi_y)^{1 - y^{(i)}}$
- $p(x_j^{(i)} | y^{(i)} = b, \phi_{j|y=b}) = \phi_{j|y=b}^{x_j^{(i)}} (1 - \phi_{j|y=b})^{1 - x_j^{(i)}}$

Thus, the joint probability can be written as:

$$p(x^{(i)}, y^{(i)} | \phi_y, \phi_{j|y=b}) = \phi_y^{y^{(i)}} (1 - \phi_y)^{1 - y^{(i)}} \cdot \prod_{j=1}^{n} \phi_{j|y=y^{(i)}}^{x_j^{(i)}} (1 - \phi_{j|y=y^{(i)}})^{1 - x_j^{(i)}}$$

The likelihood function for the entire dataset is the product of the joint probabilities for all data points:

$$L(\phi_y, \phi_{j|y=b}) = \prod_{i=1}^{m} p(x^{(i)}, y^{(i)} | \phi_y, \phi_{j|y=b}) = \prod_{i=1}^{m} \left(\phi_y^{y^{(i)}} (1 - \phi_y)^{1 - y^{(i)}} \prod_{j=1}^{n} \phi_{j|y=y^{(i)}}^{x_j^{(i)}} (1 - \phi_{j|y=y^{(i)}})^{1 - x_j^{(i)}} \right)$$

To simplify calculations, we take the logarithm of the likelihood function:

$$\log L(\phi_y, \phi_{j|y=b}) = \sum_{i=1}^{m} \left(y^{(i)} \log \phi_y + (1 - y^{(i)}) \log(1 - \phi_y) \right)$$

$$+ \sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_j^{(i)} \log \phi_{j|y=y^{(i)}} + (1 - x_j^{(i)}) \log(1 - \phi_{j|y=y^{(i)}}) \right)$$

We now compute the MLE of the parameters by taking the derivative of the log-likelihood function and setting it to zero.

For ϕ_y:

$$\frac{\partial \log L}{\partial \phi_y} = \sum_{i=1}^{m} \left(\frac{y^{(i)}}{\phi_y} - \frac{1 - y^{(i)}}{1 - \phi_y} \right)$$

Setting this to zero, we obtain the MLE for ϕ_y:

$$\hat{\phi}_y = \frac{1}{m} \sum_{i=1}^{m} y^{(i)}$$

This is the proportion of samples where $y = 1$.

For $\phi_{j|y=b}$:

$$\frac{\partial \log L}{\partial \phi_{j|y=b}} = \sum_{i=1}^{m} \left(\frac{x_j^{(i)}}{\phi_{j|y=b}} - \frac{1 - x_j^{(i)}}{1 - \phi_{j|y=b}} \right) 1\{y^{(i)} = b\}$$

$$\hat{\phi}_{j|y=b} = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = b\} x_j^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = b\}}$$

This is the proportion of times feature $x_j = 1$ when $y = b$.

The maximum likelihood estimates for the parameters are:

$$\hat{\phi}_y = \frac{1}{m} \sum_{i=1}^{m} y^{(i)}$$

$$\hat{\phi}_{j|y=b} = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = b\} x_j^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = b\}}$$

Problem 2.3: Comparison of Genetrative and Discriminative Models

(a) Parameter Estimation for GDA

We are asked to derive the parameter estimation for Gaussian Discriminant Analysis (GDA) with a shared covariance matrix. The setup is as follows:

- $y \sim \text{Bernoulli}(\phi)$
- $x|y = 0 \sim \mathcal{N}(\mu_0, \Sigma)$
- $x|y = 1 \sim \mathcal{N}(\mu_1, \Sigma)$

To estimate the parameters μ_0, μ_1, Σ, and ϕ, we use the maximum likelihood method.

- **Estimate** ϕ: The probability of the label $y = 1$ is estimated by the fraction of positive labels:
$$\hat{\phi} = \frac{1}{n}\sum_{i=1}^{n} y^{(i)}$$
- **Estimate** μ_0 **and** μ_1: The mean vectors μ_0 and μ_1 are the empirical means of the samples that have $y = 0$ and $y = 1$ respectively:
$$\hat{\mu}_0 = \frac{\sum_{i:y^{(i)}=0} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = 0\}}$$
$$\hat{\mu}_1 = \frac{\sum_{i:y^{(i)}=1} x^{(i)}}{\sum_{i=1}^{n} 1\{y^{(i)} = 1\}}$$
- **Estimate** Σ: The shared covariance matrix Σ is estimated by pooling the covariance across both classes:
$$\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} \left(x^{(i)} - \mu_{y^{(i)}}\right)\left(x^{(i)} - \mu_{y^{(i)}}\right)^T$$
where $\mu_{y^{(i)}}$ is μ_0 if $y^{(i)} = 0$, and μ_1 if $y^{(i)} = 1$.

(b) Compute for the Parameters

The dataset provided is:

$$x^{(1)} = \begin{pmatrix}1\\2\end{pmatrix}, \quad y^{(1)} = 0$$
$$x^{(2)} = \begin{pmatrix}2\\3\end{pmatrix}, \quad y^{(2)} = 0$$
$$x^{(3)} = \begin{pmatrix}3\\4\end{pmatrix}, \quad y^{(3)} = 1$$
$$x^{(4)} = \begin{pmatrix}4\\5\end{pmatrix}, \quad y^{(4)} = 1$$

- **Estimate** ϕ:

$$\hat{\phi} = \frac{1}{4}\left(y^{(1)} + y^{(2)} + y^{(3)} + y^{(4)}\right) = \frac{1}{4}(0+0+1+1) = \frac{1}{2}$$

- **Estimate** μ_0:

$$\hat{\mu}_0 = \frac{x^{(1)} + x^{(2)}}{2} = \frac{1}{2}\left(\begin{pmatrix}1\\2\end{pmatrix} + \begin{pmatrix}2\\3\end{pmatrix}\right) = \begin{pmatrix}1.5\\2.5\end{pmatrix}$$

- **Estimate** μ_1:

$$\hat{\mu}_1 = \frac{x^{(3)} + x^{(4)}}{2} = \frac{1}{2}\left(\begin{pmatrix}3\\4\end{pmatrix} + \begin{pmatrix}4\\5\end{pmatrix}\right) = \begin{pmatrix}3.5\\4.5\end{pmatrix}$$

- **Estimate** Σ: First, we calculate the covariance terms. For the class $y = 0$:

$$(x^{(1)} - \mu_0) = \begin{pmatrix}1\\2\end{pmatrix} - \begin{pmatrix}1.5\\2.5\end{pmatrix} = \begin{pmatrix}-0.5\\-0.5\end{pmatrix}$$

$$(x^{(2)} - \mu_0) = \begin{pmatrix}2\\3\end{pmatrix} - \begin{pmatrix}1.5\\2.5\end{pmatrix} = \begin{pmatrix}0.5\\0.5\end{pmatrix}$$

  For the class $y = 1$:

$$(x^{(3)} - \mu_1) = \begin{pmatrix}3\\4\end{pmatrix} - \begin{pmatrix}3.5\\4.5\end{pmatrix} = \begin{pmatrix}-0.5\\-0.5\end{pmatrix}$$

$$(x^{(4)} - \mu_1) = \begin{pmatrix}4\\5\end{pmatrix} - \begin{pmatrix}3.5\\4.5\end{pmatrix} = \begin{pmatrix}0.5\\0.5\end{pmatrix}$$

  Now calculate the covariance matrix:

$$\Sigma = \frac{1}{4}\left(\begin{pmatrix}-0.5\\-0.5\end{pmatrix}\begin{pmatrix}-0.5 & -0.5\end{pmatrix} + \begin{pmatrix}0.5\\0.5\end{pmatrix}\begin{pmatrix}0.5 & 0.5\end{pmatrix} + \begin{pmatrix}-0.5\\-0.5\end{pmatrix}\begin{pmatrix}-0.5 & -0.5\end{pmatrix} + \begin{pmatrix}0.5\\0.5\end{pmatrix}\begin{pmatrix}0.5 & 0.5\end{pmatrix}\right)$$

$$= \begin{pmatrix}0.25 & 0.25\\0.25 & 0.25\end{pmatrix}$$

The computed parameters for this dataset are:

- $\hat{\phi} = 0.5$
- $\hat{\mu}_0 = \begin{pmatrix}1.5\\2.5\end{pmatrix}$
- $\hat{\mu}_1 = \begin{pmatrix}3.5\\4.5\end{pmatrix}$
- $\hat{\Sigma} = \begin{pmatrix}0.25 & 0.25\\0.25 & 0.25\end{pmatrix}$

(c) Decision Boundary for LDA

The decision boundary is derived by finding the point where the probabilities of the two classes are equal, i.e., $P(y=0|x) = P(y=1|x)$.Since both classes follow Gaussian distributions with equal covariance matrices, the decision rule simplifies to:

$$\hat{y} = \hat{y} = \arg\max_y P(y|x) = \arg\max_y \log P(x|y) + \log P(y)$$

This leads to the decision boundary equation:

$$x^T\Sigma^{-1}(\mu_0 - \mu_1) = \frac{1}{2}(\mu_0^T\Sigma^{-1}\mu_0 - \mu_1^T\Sigma^{-1}\mu_1) + \log\left(\frac{\phi}{1-\phi}\right)$$

(d) Decision Boundaries Comparion of LDA and LR

a. LDA Decision Boundary

From the part (c),we derived that the decision boundary for LDA is given by:

$$x^T\Sigma^{-1}(\mu_0 - \mu_1) = \frac{1}{2}(\mu_0^T\Sigma^{-1}\mu_0 - \mu_1^T\Sigma^{-1}\mu_1) + \log\left(\frac{\phi}{1-\phi}\right)$$

When $\Sigma = I$(Identity Mrix),the decision boundary simplifies to:

$$x^T(\mu_0 - \mu_1) = \frac{1}{2}(\mu_0^T\mu_0 - \mu_1^T\mu_1) + \log\left(\frac{\phi}{1-\phi}\right)$$

b. LR Decision Boundary

Logistic regression models the conditional probability $P(y=1|x)$ directly using a sigmoid function:

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

The decision boundary is found by solving:

$$P(y=1|x) = P(y=0|x)$$

or equivalently:

$$\theta^T x = 0$$

c. Comparison

- Both LDA (with $\Sigma = I$) and LR have **linear decision boundaries**.
- LDA assumes a Gaussian distribution for the data (generative approach), while Logistic Regression does not make any distributional assumptions and models the posterior probabilities directly (discriminative approach).
- If $\Sigma \neq I$, the decision boundary for LDA can still be linear but may become more complex depending on the covariance structure, whereas Logistic Regression will always produce a linear boundary.

(e) LDA vs LR on Small Datasets

a. LR on Small Datasets

- Logistic Regr ession is a **discriminative model**, which means it directly models $P(y|x)$ without making assumptions about the underlying distribution of the data.
- On small datasets, this can be problematic because logistic regression might not have enough data to accurately estimate $P(y|x)$ without overfitting. It lacks the **regularizing effect** of modeling the data's distribution, which a generative model provides.

b. LDA on Small Datasets

- LDA is a **generative model**, meaning it models the joint distribution $P(x, y)$, specifically $P(x|y)$ and $P(y)$.
- LDA benefits from making stronger assumptions about the data (e.g., Gaussian distribution with shared covariance).

c. Why Generative Models Might Perform Better

- **With small datasets**, the limited amount of data can make it difficult for discriminative models like logistic regression to accurately learn the conditional probabilities.
- **Generative models**, such as LDA, can perform better because they make use of the **entire data distribution**, imposing structure via assumptions like Gaussianity. This helps stabilize predictions, even with small data.