Robust logistic regression with shift parameter estimation (1-18).

Prerequisites:
Logistic Regression
Empirical Risk Minimization
Margin in Classification
Robust Statistics
Optimization Techniques
Penalty Functions
Statistical Learning Theory
Basic Probability and Statistics

1. **Logistic Regression:**

Logistic Regression is a statistical method used for **binary classification**, where the goal is to model the relationship between a set of independent variables (features) and a dependent variable (outcome) that takes binary values (e.g., 0/1 or Yes/No). Logistic regression predicts the **probability** of an outcome belonging to one of two classes. Logistic regression maps the output using the **logistic function (sigmoid function)** to constrain the predicted probabilities to the range [0,1].

$$P(y=1|X) = \frac{1}{1+e^{-z}},\ where\ z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p$$

The predicted probability is compared to a threshold (commonly $t = 0.5$) to classify the instance as $y = 1 \ \ if \ \ P \geq 0.5$. Otherwise $y = 0$.

Loss Function: Logistic regression uses the **log-loss (negative log-likelihood)** as the optimization criterion.

$$R(f) = -\frac{1}{n}\sum_{i=1}^{n}[\ y_i\ ln(\hat{y}_i)\ +\ (1-y_i)\ ln(1-\hat{y}_i)]$$

y_i is the actual label and $\hat{y}_i$ is the predicted probability of $y_i = 1$

Optimization: The coefficients β are learned by maximizing the likelihood of observing the data (or equivalently minimizing the log-loss) using methods like **Gradient Descent**.

2. **Empirical Risk Minimization:**

Empirical Risk is a concept in statistical learning and machine learning that refers to the **average loss** incurred on a given dataset when using a specific model. It represents how well the model performs on the training data and is used to evaluate and optimize the model during the training process. Empirical risk is mathematically expressed as the average of the **loss function** over all training examples:

$$R(f) = \frac{1}{n}\sum_{i=1}^{n} L(y_i,\ f(x_i))$$

The loss function defines the penalty for incorrect predictions.

$$If,\ y \in \{-1, +1\},\ then\ log(1 + exp(-yf(x)))$$

$$If,\ y \in \{0, +1\},\ then\ \ -y\ ln(\hat{y}) - (1 - y)\ ln(1 - \hat{y})$$

Introduction and Motivation:
This section of the research paper focuses on the challenges of logistic regression in the presence of **outliers** and **label noise** and proposes a solution using a **shift parameter approach** to improve robustness.

Robustness: refers to the ability of a model, algorithm, or statistical method to maintain **accuracy, stability, and reliability** even when the assumptions about the data are violated. This typically means being resistant to **outliers, noise, model misspecifications, or adversarial changes** in the dataset.

Outliers are data points that deviate significantly from the expected distribution of the data. They can arise due to:

Noise in input features: Extreme values in the feature space x_i.

Label Noise: Incorrect or flipped class labels y_i.

Outliers affect logistic regression because logistic loss is
unbounded. A single severe outlier with a large negative margin can disproportionately increase the loss.

Unbounded: Logistic loss $log(1 + exp(-u)),\ where\ u = y.f(x)$ is unbounded because;
For
correct classifications - (u>0): The loss decreases as u increases, approaching 0 for large u.
For
misclassifications - **(u<0):** Loss grows exponentially as u→−∞, meaning outliers with large negative margins can cause the loss to blow up, which can dominate during model training.

This sensitivity to outliers arises because the logistic loss does not cap the penalty for large negative margins, unlike bounded loss functions. Hence, outliers with extreme values heavily influence logistic regression, necessitating robust methods like the shift parameter approach to mitigate this issue.

Existing Solutions and their Challenges:
Bounded Loss Functions:
Non-convex and computationally expensive.
Label Noise Modeling - Requires knowledge of the label transition probabilities, which are often unknown in practice.

Margin in Classification: Measure of how confidently a model assigns a prediction to a particular class. It quantifies the distance between a data point and the decision boundary (or hyperplane) in the feature space.

$u = y.f(x)$.

$u > 0$ - Classifier predicted correctly.

$u < 0$ - Classifier predicted incorrectly.

$u = 0$ - Classifier is uncertain.

Decision Boundary: The surface or line that separates different classes in the feature space based on the decision boundary $f(x)$. It is where the classifier is indifferent between classes, i.e., where $f(x) = 0$.

Proposed Solution: Shift Parameter Approach

To address these issues, the authors propose adding **shift parameters** γ_i for each observation to adjust the margins:

$$R(f, \gamma_i) = \sum_{i=1}^{n} [\, log(1 + exp(-y_i\, f(x_i) + \gamma_i)) + p_\gamma(\gamma_i)]$$

where:

γ_i : A shift parameter for the i-th observation, which adjusts the margin.

$p_\gamma(\gamma_i)$: A regularization penalty function that encourages sparsity in γ_i (most γ_i should be 0).

Shift Parameters:
Are additional case-specific (observation-level) parameters introduced into the logistic regression framework to make the classification process
robust against outliers and label noise.
The shift parameter
γ_i modifies the margin $u = y_i f(x_i)$ to an adjusted margin.

$$u^{'} = y_i f(x_i) - \gamma_i$$

How it works: For misclassified points $[(y_i f(x_i)) < 0]$ the shift parameter $\gamma_i < 0$ adjusts the margin upward, reducing the impact of severe outliers.

Penalty Term: $p_\lambda(\gamma_i)$ is applied to prevent the shift parameters from arbitrarily large adjustments.
and also encourage sparsity.

Why penalty functions?

If unconstrained, large negative margins may cause γ_i become excessively large.
Penalty functions ensure only necessary adjustments (dynamically) are made while keeping
γ_i small.
The penalty encourages sparsity, meaning that most observations will have
$\gamma_i = 0$, and only a few outliers will be adjusted.
Practical Impact of Choosing the Right Penalty:

Penalty Function	Encourages Sparsity?	Allows Large Adjustments?	Best For
L1 (Lasso)	Yes	No	Small outliers, sparse corrections
SCAD (Smoothly Clipped Absolute Deviation)	Partial	Moderate	General robust classification
MCP (Minmax Concave Penalty)	Strong	Yes	Large outliers, extreme corrections

Optimization with Shift Parameters:

Step-1 Classifier Update: Fix γ_i and solve for $f(x)$ → Solved used Gradient Descent for adjusted margins.

Step-2 Shift Parameter Update: Involves solving for γ_i, →

$\gamma_i = arg\, min_\gamma [log(1 + exp(-y_i f(x_i) + \gamma)) + p_\lambda(\gamma)]$.
* Can be solved in
closed form for L1 Regularization.

*

For SCAD/MCP, requires an **iterative thresholding method**.

-Refer

Inspiration from wavelet denoising:

The idea of shift parameters is inspired by
wavelet thresholding which are used to suppress noise in signals.

1. Soft Thresholding: $\hat{w} = sign(w)\ max(|w| - \lambda,\ 0)$, Shrinks small values towards zero.

2. Hard Thresholding: $\hat{w} = \begin{cases} w, & |w| > \lambda \\ 0, & |w| \leq \lambda \end{cases}$, Completely removes below a threshold.

Methodology:

This paper showcases:

-

The **empirical risk minimization with shift parameters** leads to better robustness.

- The

alternating optimization converges under mild conditions.

- The

penalty functions ensure sparsity, preventing overfitting.

Methodology Summary:

Step	Mathematical Update	Purpose
Initialize $f(x)$, set $\gamma_i = 0$	-	Start with normal logistic regression.
Update $f(x)$ (Fix γ_i)	$min_f \sum_i log(1 + exp(-y_i f(x_i) + \gamma_i))$	Learn Classifier
Update γ_i (Fix $f(x)$)	$\gamma_i = arg\ min_\gamma [log(1 + exp(-y_i f(x_i) + \gamma) + p_\lambda(\gamma))]$	Adjust Outliers
Repeat Until Convergence	Alternating Minimization	Ensure robustness

Computational Complexity:

- The 2 update iteration optimization approach scales well compared to traditional methods that require non-convex optimization.

-END-
