Stepwise Forward Selection for Logistic Regression

Mathematical Explanation and Implementation

May 4, 2025

1 Introduction

Stepwise forward selection is a greedy feature selection algorithm that iteratively adds features to a model based on their contribution to model performance. In this document, we explain the mathematical reasoning behind the stepwise forward selection algorithm and its implementation for logistic regression on the heart disease dataset.

2 Mathematical Framework

2.1 Logistic Regression

Logistic regression models the probability of a binary outcome $y \in \{0, 1\}$ given features $x \in \mathbb{R}^d$ as:

$$P(y = 1|x) = \sigma(w \cdot x + b) = \frac{1}{1 + e^{-(w \cdot x + b)}} \tag{1}$$

where σ is the sigmoid function, $w \in \mathbb{R}^d$ is the weight vector, and $b \in \mathbb{R}$ is the bias term.
The log-likelihood function for a dataset $\{(x^{(i)}, y^{(i)})\}_{i=1}^n$ is:

$$\mathcal{L}(w, b) = \sum_{i=1}^{n} \left[y^{(i)} \log(\sigma(w \cdot x^{(i)} + b)) + (1 - y^{(i)}) \log(1 - \sigma(w \cdot x^{(i)} + b)) \right] \tag{2}$$

In logistic regression, we find the parameters w and b that maximize this log-likelihood.

2.2 Feature Selection Problem

In many applications, we want to use only a subset of the available features to:

- Reduce model complexity
- Improve interpretability
- Avoid overfitting
- Reduce computational cost

Given d features, there are 2^d possible feature subsets, making exhaustive search impractical for large d. Stepwise forward selection provides a greedy approach to this problem.

2.3 Stepwise Forward Selection Algorithm

The stepwise forward selection algorithm for selecting k features works as follows:

Stepwise Forward Selection Algorithm

1. Initialize $S = \emptyset$ (empty set of selected features)
2. For $j = 1$ to k:
   (a) Set best_error = ∞
   (b) Set best_feature = None
   (c) For each feature $f \notin S$:
      i. Estimate the error of a model using features $S \cup \{f\}$
      ii. If error $<$ best_error:
         A. best_error = error
         B. best_feature = f
   (d) $S = S \cup \{\text{best\_feature}\}$
3. Train final model using only features in S
4. Return S and the trained model

2.4 Error Estimation via Cross-Validation

To estimate the error of a model with a given feature subset, we use K-fold cross-validation:

1. Divide the training data into K equal-sized folds
2. For each fold i:
   - Train the model on all folds except fold i
   - Test the model on fold i and compute the error
3. Average the errors across all K folds

The cross-validation error is given by:

$$\text{CV Error} = \frac{1}{K} \sum_{i=1}^{K} \text{Error}_i \tag{3}$$

where Error_i is the error on fold i.

3 Implementation Details

3.1 Error Estimation Function

The $error_estimate(X, y, S)$ function:

Error Estimation Function

1. Takes a dataset (X, y) and a set of feature indices S
2. Selects only the features in S from X
3. Creates a logistic regression model without regularization (penalty=None)
4. Performs 5-fold cross-validation
5. Returns the error rate (1 - accuracy)

Mathematically, for each fold i, we:

1. Train a logistic regression model on the training portion of fold i, using only features in S
2. Compute the accuracy on the validation portion of fold i
3. Average the accuracies across all folds
4. Return 1 - average accuracy as the error estimate

3.2 Stepwise Forward Selection Function

The $stepwise_forward_selection(X_train, y_train, k)$ function:

Stepwise Forward Selection Function

1. Initializes an empty set S of selected features
2. For each iteration (up to k):
   (a) For each feature f not in S, estimates the error of a model using features $S \cup \{f\}$
   (b) Adds the feature that results in the lowest error to S
3. Returns the set S of selected features

3.3 Model Evaluation

For each value of k from 1 to 13:

1. We select the k best features using stepwise forward selection
2. Train a logistic regression model on the training data using only these features
3. Compute the cross-validation error on the training data
4. Compute the test error on the held-out test data
5. Plot both errors against k

3.4 Decision Boundary Visualization

For $k = 2$, we:

1. Identify the two selected features
2. Train a logistic regression model using only these two features
3. Create a meshgrid covering the feature space
4. Predict the class for each point in the meshgrid
5. Visualize the decision boundary along with the training data points

The decision boundary is the set of points x where $P(y = 1|x) = 0.5$, which corresponds to $w \cdot x + b = 0$.

4 Mathematical Justification

4.1 Why Stepwise Forward Selection Works

Stepwise forward selection is a greedy algorithm that makes locally optimal choices at each step. While it doesn't guarantee finding the globally optimal feature subset, it often works well in practice because:

1. It considers the interaction between features already selected and new candidate features
2. It directly optimizes the performance metric of interest (classification error)
3. It avoids the combinatorial explosion of exhaustive search

4.2 Limitations

The main limitations of stepwise forward selection are:

1. It may not find the globally optimal feature subset due to its greedy nature
2. It doesn't account for redundancy among features (unlike methods like LASSO)
3. It can be computationally expensive for large datasets with many features
4. It may be sensitive to noise in the data

4.3 Comparison with L1 Regularization

An alternative approach to feature selection is L1 regularization (LASSO), which adds a penalty term proportional to the L1 norm of the weight vector:

$$\mathcal{L}_{\text{LASSO}}(w, b) = \mathcal{L}(w, b) - \lambda \|w\|_1 \tag{4}$$

The key differences between stepwise forward selection and L1 regularization are:

1. L1 regularization performs feature selection and model fitting simultaneously
2. L1 regularization can handle correlated features better
3. Stepwise forward selection directly optimizes the performance metric
4. Stepwise forward selection gives more control over the exact number of features

5 Conclusion

Stepwise forward selection provides a practical approach to feature selection for logistic regression. By iteratively adding features that most improve model performance, it creates sparse models that can be more interpretable and potentially generalize better than models using all available features.

The implementation demonstrated in this document applies stepwise forward selection to the heart disease dataset, evaluating models with different numbers of features and visualizing the decision boundary for the two-feature model.