Lasso Regression for House Price Prediction

Your Name

April 17, 2025

Abstract

In this project, we investigate the application of Lasso regression for predicting house prices using the Boston Housing dataset. Lasso, or Least Absolute Shrinkage and Selection Operator, is a linear regression technique that includes an L_1 regularization term. This allows for automatic feature selection by shrinking some coefficients to zero, which improves model interpretability and can help with generalization. We provide the mathematical formulation, implement the method using Python, and evaluate its performance.

1 Introduction

House price prediction is a classic regression problem in machine learning and real estate analytics. The goal is to predict the market value of a house given features such as location, number of rooms, and crime rate.

While ordinary least squares regression can be used, it may lead to overfitting in the presence of multicollinearity or irrelevant features. Lasso regression addresses this by adding an L_1 penalty to the loss function, encouraging sparsity in the coefficients. This makes Lasso a valuable tool for high-dimensional problems and for understanding which features are most relevant.

2 Mathematical Formulation of Lasso

Let $A \in \mathbb{R}^{m \times n}$ be the feature matrix, and $b \in \mathbb{R}^m$ be the target vector (house prices). The Lasso regression problem is defined as:

$$\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1 \right\}$$

where:

- $\|Ax - b\|_2^2$ is the least squares loss,
- $\|x\|_1 = \sum_{i=1}^{n} |x_i|$ is the L_1 norm of the coefficients,
- $\lambda > 0$ is the regularization parameter that controls sparsity.

The Lasso problem is convex but not differentiable due to the L_1 norm. Efficient algorithms such as coordinate descent or proximal gradient methods are commonly used for optimization.

3 Python Implementation

We used the Boston Housing dataset available on Kaggle. The following code snippet demonstrates how to apply Lasso regression using `scikit-learn`:

Loading the Dataset and Preprocessing

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import Lasso
from sklearn.metrics import mean_squared_error

# Load data (ensure 'boston.csv' is downloaded from Kaggle)
data = pd.read_csv("boston.csv")

# Features and target
X = data.drop(columns=["MEDV"])  # MEDV is the target: median value
y = data["MEDV"]

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
   =0.2, random_state=42)

# Standardize features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Fitting the Lasso Model

```
# Fit Lasso model
lasso = Lasso(alpha=1.0)
lasso.fit(X_train_scaled, y_train)

# Predictions and evaluation
y_pred = lasso.predict(X_test_scaled)
mse = mean_squared_error(y_test, y_pred)
print(f"Test MSE: {mse:.2f}")
```

4 Conclusion

Lasso regression is an effective method for both prediction and feature selection in regression tasks. Applied to the Boston Housing dataset, it can reduce model complexity while maintaining good predictive performance. Tuning the regularization parameter λ is essential for balancing bias and variance. In practice, Lasso can help identify which housing features are truly important in determining price.