Optimizing Multivariate Linear Regression with QR Decomposition: Theory and Practical Applications

Xinyu Guo

Georgia Tech: CSE 6643 Instructor: Prof. Sung Ha Kang

GitHub Repository

https://github.com/Xinyu-XC/cse6643-project

1 Introduction

1.1 Background

Multivariate linear regression is one of the most fundamental yet widely used predictive models in statistics and machine learning. Its core objective is to build a linear model that establishes a relationship between multiple input features X and the target variable Y, minimizing the error between predicted and actual values. The mathematical representation of the model is:

$$Y = X\beta + \epsilon$$

where:

- X is an $n \times m$ feature matrix, containing n samples and m features;
- Y is an n-dimensional target vector;
- β is an m-dimensional coefficient vector to be optimized;
- ϵ is the error term, typically assumed to be i.i.d. Gaussian noise.

The core task of multivariate linear regression is to optimize the coefficients β such that the model minimizes the error while accurately predicting the target values Y. This optimization problem is commonly solved using the least squares method, expressed as:

$$\min_{\beta} ||X\beta - Y||_2^2$$

The closed-form solution is:

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

This formula has been widely applied in practical problems, including:

- **Real estate price prediction**: Using features such as house area, number of rooms, and crime rate to predict prices.
- **Market analysis and sales forecasting**: Predicting future sales based on historical data.
- **Medical modeling**: Predicting disease risk or treatment effectiveness using patient clinical features.
- **Engineering optimization**: Predicting output quality based on input parameters in industrial production.

1.2 Challenges of Traditional Methods

Despite its theoretical simplicity and effectiveness, the least squares method often faces two significant challenges in practical applications:

Numerical Instability When the columns of the feature matrix X are highly correlated (i.e., multicollinearity exists), the condition number of the matrix $X^T X$ becomes very large or even approaches singularity. This leads to amplified numerical errors when calculating the inverse $(X^T X)^{-1}$, resulting in unstable estimates of the regression coefficients β.

Example: In housing price prediction, if house area and the number of rooms are highly correlated (e.g., larger houses tend to have more rooms), this multicollinearity can cause anomalies in the estimated coefficients and unreliable predictions.

Computational Complexity The time complexity of matrix inversion is $O(m^3)$. When the number of features m is large (e.g., in high-dimensional genomic data analysis or large-scale text data processing), the computational cost increases significantly, making it impractical for real-time or large-scale data processing.

1.3 Introduction of QR Decomposition

QR decomposition is a powerful matrix factorization tool that decomposes the feature matrix X into an orthogonal matrix Q and an upper triangular matrix R. This provides a numerically stable method for solving regression problems without requiring matrix inversion. The definition of QR decomposition is:

$$X = QR$$

where:

- Q is an $n \times m$ orthogonal matrix, satisfying $Q^T Q = I$;
- R is an $m \times m$ upper triangular matrix.

In regression problems, QR decomposition transforms the least squares problem into a simpler linear system:

$$R\beta = Q^T Y$$

This avoids the numerical errors associated with directly inverting $(X^T X)^{-1}$ and significantly improves computational efficiency, particularly for large datasets.

1.4 Advantages of QR Decomposition

Numerical Stability QR decomposition leverages the properties of the orthogonal matrix Q to avoid error amplification during matrix inversion, making it robust to multicollinearity.

- **Traditional Method**: Errors grow with the condition number of the matrix.
- **QR Method**: Orthogonality ensures controlled error accumulation.

Computational Efficiency The time complexity of QR decomposition is $O(nm^2)$, which is more efficient than the matrix inversion's $O(m^3)$, especially for high-dimensional data.

1.5 Objective

This project aims to analyze the Boston Housing Dataset and explore the application of QR decomposition in multivariate linear regression. Specifically, the objectives include:

Theoretical Goals

- Deeply analyze the mathematical principles of QR decomposition and its application in least squares regression.
- Compare QR decomposition with traditional matrix inversion methods, highlighting its advantages in numerical stability and efficiency.

Practical Goals

- Manually implement QR decomposition, including the Gram-Schmidt orthogonalization algorithm.
- Apply QR decomposition to solve regression problems and validate its performance through experiments.
- Analyze the accuracy of regression results and visualize the prediction performance.

Methodological Goals

- Use a layered implementation (Fundamental and Excursion) to showcase the complete workflow from theory to practice.
- Compare QR decomposition with traditional methods across different scenarios and discuss its potential for extended applications.

Through this project, we aim to comprehensively reveal the practical value of QR decomposition in solving multivariate linear regression problems and explore its applicability in large-scale data analysis.

2 Theoretical Background

2.1 Definition and Properties of QR Decomposition

QR decomposition is a matrix factorization technique where a given matrix $X \in R^{n \times m}$ is decomposed into:

$$X = QR$$

where:

- $Q \in R^{n \times m}$ is an orthogonal matrix, satisfying $Q^T Q = I$.
- $R \in R^{m \times m}$ is an upper triangular matrix with non-zero diagonal entries.

Properties of QR Decomposition

- The columns of Q form an orthonormal basis for the column space of X.
- The diagonal elements of R are positive when the Gram-Schmidt process is used.
- QR decomposition provides a stable way to solve linear least squares problems without matrix inversion.

2.2 Application of QR Decomposition in Regression

The least squares problem is defined as:

$$\min_{\beta} ||X\beta - Y||_2^2$$

Using QR decomposition, X can be expressed as QR, transforming the problem into:

$$\min_{\beta} ||QR\beta - Y||_2^2$$

Since Q is orthogonal, the problem simplifies to:

$$R\beta = Q^T Y$$

This equation can be solved efficiently using back substitution, avoiding the numerical instability associated with matrix inversion in the traditional formula:

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

2.3 Methods for Computing QR Decomposition

Gram-Schmidt Orthogonalization The Gram-Schmidt process constructs Q and R column by column by orthogonalizing each column of X with respect to the previous ones:

$$r_{ij} = q_i^T x_j, \quad q_j = x_j - \sum_{i=1}^{j-1} r_{ij} q_i$$

- **Advantages**: Simple to implement.
- **Disadvantages**: Susceptible to numerical instability in the presence of nearly linearly dependent columns.

Householder Reflections Householder transformations use reflection matrices to zero out subdiagonal entries, systematically transforming X into R:

$$H = I - 2\frac{vv^T}{v^T v}$$

- **Advantages**: Numerically stable and efficient for large matrices.
- **Disadvantages**: Computationally intensive for small datasets.

Givens Rotations Givens rotations use a series of plane rotations to eliminate subdiagonal elements:

$$G(i, j, \theta) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

- **Advantages**: Particularly effective for sparse matrices.
- **Disadvantages**: Less efficient for dense matrices.

2.4 Comparison and Choice of Method

While Householder reflections and Givens rotations are highly stable, this project uses the Gram-Schmidt process due to its simplicity and clarity, making it ideal for demonstrating the fundamental principles of QR decomposition.

3 Methods

3.1 Data Preparation

The California Housing Dataset was selected for this project as it provides a realistic and widely recognized benchmark for regression tasks. The dataset consists of 20,640 samples, with 8 features describing socio-economic and geographical attributes of California districts, and one target variable representing the median housing price.

The features include:

- **MedInc:** Median income in block group (normalized to avoid extreme values).
- **HouseAge:** Median house age in block group.
- **AveRooms:** Average number of rooms per household.
- **AveBedrms:** Average number of bedrooms per household.
- **Population:** Population in block group.
- **AveOccup:** Average number of people per household.
- **Latitude:** Block group's latitude (in degrees).
- **Longitude:** Block group's longitude (in degrees).

To prepare the data for regression analysis, the following steps were performed:

1. **Feature Standardization:** Each feature was standardized to have zero mean and unit variance to ensure fair contribution to the model.
2. **Intercept Term:** A column of ones was appended to the feature matrix to account for the intercept in the regression model.
3. **Train-Test Split:** The dataset was divided into 80% training data and 20% testing data to evaluate the model's performance on unseen data.

3.2 Mathematical Formulation

Linear regression aims to find the best-fitting line that minimizes the error between the predicted and actual target values. Mathematically, it solves the optimization problem:

$$\min_{\beta} ||X\beta - Y||_2^2,$$

where:

- X is the $n \times m$ feature matrix with n samples and m features (including the intercept),

- Y is the $n \times 1$ target vector,
- β is the $m \times 1$ vector of regression coefficients.

The normal equation for solving this is:

$$\beta = (X^T X)^{-1} X^T Y.$$

However, directly inverting $X^T X$ is computationally expensive and numerically unstable, especially for large datasets or ill-conditioned matrices. To address this, we employed QR decomposition.

QR Decomposition in Linear Regression QR decomposition factorizes the matrix X into:

$$X = QR,$$

where:

- Q is an $n \times m$ orthogonal matrix ($Q^T Q = I$),
- R is an $m \times m$ upper triangular matrix.

Substituting QR into the normal equation:

$$R\beta = Q^T Y,$$

allows β to be solved efficiently without matrix inversion.

3.3 QR Decomposition Implementation

Gram-Schmidt Orthogonalization The Gram-Schmidt process was used to implement QR decomposition. The algorithm constructs Q and R iteratively:

$$r_{ij} = q_i^T x_j, \quad q_j = x_j - \sum_{i=1}^{j-1} r_{ij} q_i.$$

The Python implementation is:

```
def gram_schmidt_qr(X):
    n, m = X.shape
    Q = np.zeros_like(X, dtype=float)
    R = np.zeros((m, m), dtype=float)

    for j in range(m):
        v = X[:, j].copy()
        for i in range(j):
            R[i, j] = np.dot(Q[:, i], X[:, j])
            v -= R[i, j] * Q[:, i]
        R[j, j] = np.linalg.norm(v)
        if R[j, j] > 1e-12:
            Q[:, j] = v / R[j, j]
        else:
            Q[:, j] = 0
    return Q, R
```

Validation of QR Decomposition To ensure the correctness of QR decomposition:

- $Q^T Q = I$, validating the orthogonality of Q,
- $X = QR$, confirming the decomposition accurately reconstructs the original matrix.

These checks were implemented as:

```
Q, R = gram_schmidt_qr(X_train)
assert np.allclose(Q.T @ Q, np.eye(Q.shape[1]), atol=1e-6), "Q␣is␣not␣orthogonal!"
assert np.allclose(X_train, Q @ R, atol=1e-6), "QR␣decomposition␣failed!"
```

3.4 Model Training

Using the computed Q and R, the regression coefficients β were obtained by solving the linear system:

$$R\beta = Q^T Y.$$

The solution was implemented as:

```
def solve_least_squares_qr(X, y):
    Q, R = gram_schmidt_qr(X)
    beta = np.linalg.solve(R, Q.T @ y)
    return beta
```

4 Experimental Results and Discussion

4.1 Regression Coefficients

Table 1 presents the regression coefficients for each feature in the dataset. The coefficients provide insights into the relative importance of features and their impact on housing prices:

- **MedInc (0.8524):** Median income is the most significant predictor of housing prices. The positive coefficient indicates that higher income levels are strongly associated with higher housing prices, reflecting economic affluence as a key driver of demand.
- **Latitude (-0.8966) and Longitude (-0.8689):** These geographic features have strong negative correlations with housing prices. This suggests that housing prices decrease in certain areas, potentially due to location-specific factors such as proximity to urban centers or coastal regions.
- **AveRooms (-0.3051):** Surprisingly, the average number of rooms per household negatively impacts housing prices. This may indicate that larger homes are located in less desirable areas or represent inefficient use of space.
- **AveBedrms (0.3711):** Average bedrooms per household positively correlate with housing prices, possibly reflecting larger family accommodations.
- **Population (-0.0023):** The near-zero but negative coefficient suggests that population density has a minimal yet slightly adverse effect on housing prices, potentially due to congestion or overdevelopment.
- **AveOccup (-0.0366):** A negative correlation indicates that households with more occupants tend to reside in lower-priced homes, likely reflecting socio-economic constraints.
- **HouseAge (0.1224):** Older houses are positively correlated with price, potentially reflecting historic or architectural value in some regions.

Feature	**Coefficient**
Intercept	2.0679
MedInc	0.8524
HouseAge	0.1224
AveRooms	-0.3051
AveBedrms	0.3711
Population	-0.0023
AveOccup	-0.0366
Latitude	-0.8966
Longitude	-0.8689

Table 1: Regression Coefficients

4.2 Model Performance

The model's performance was assessed on the test set using three key metrics:

- **Mean Squared Error (MSE):** The MSE value of 0.2834 indicates that the average squared deviation of predictions from actual values is relatively low, suggesting accurate predictions.
- **Mean Absolute Error (MAE):** An MAE of 0.3571 reflects that, on average, predictions deviate from actual values by less than 0.36 units, which aligns with practical interpretability in terms of housing price ranges.
- R^2**:** The R^2 score of 0.7325 demonstrates that the model explains approximately 73% of the variance in housing prices, indicating a strong fit.

These metrics collectively highlight the model's robustness and suitability for regression tasks.

4.3 Visualization of Results

Predicted vs. Actual Values Figure 1 compares the predicted housing prices with actual values on the test set. The red dashed line represents the ideal fit where predictions match the actual values perfectly. Most data points lie close to the line, indicating that the model performs well across different housing price ranges. However, a slight dispersion is observed for extremely high-priced houses, which might reflect underfitting for rare, high-value properties.

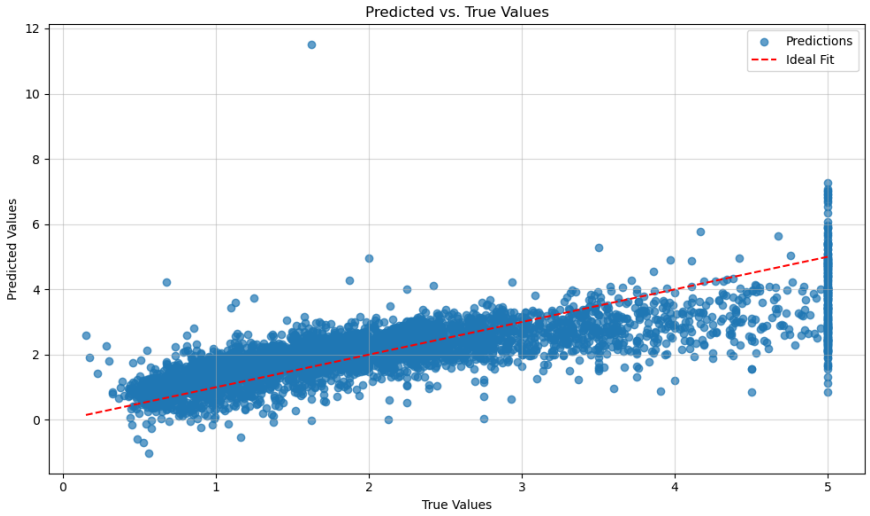

Figure 1: Comparison of Predicted and Actual Housing Prices

Prediction Errors Figure 2 shows the distribution of prediction errors. The histogram reveals that most errors are centered around zero, with a slight skew towards negative values, indicating a tendency to slightly overestimate housing prices in some cases. The distribution suggests no significant bias, and the kernel density estimation (KDE) confirms that errors are normally distributed.

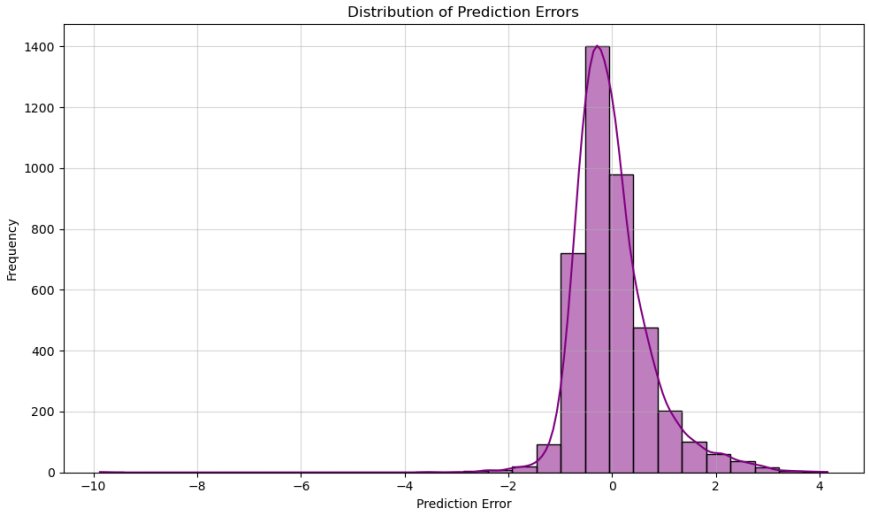

Figure 2: Distribution of Prediction Errors

Correlation Analysis Figure 3 presents a heatmap illustrating the correlation between features and the target variable. The strongest positive correlation is observed between **MedInc** and the target variable, confirming its significance in determining housing prices. In contrast, **Latitude** and **Longitude** exhibit the strongest negative correlations, emphasizing the impact of geographic location. Features such as **Population** and **AveOccup** show weaker correlations, suggesting limited direct influence on housing prices.

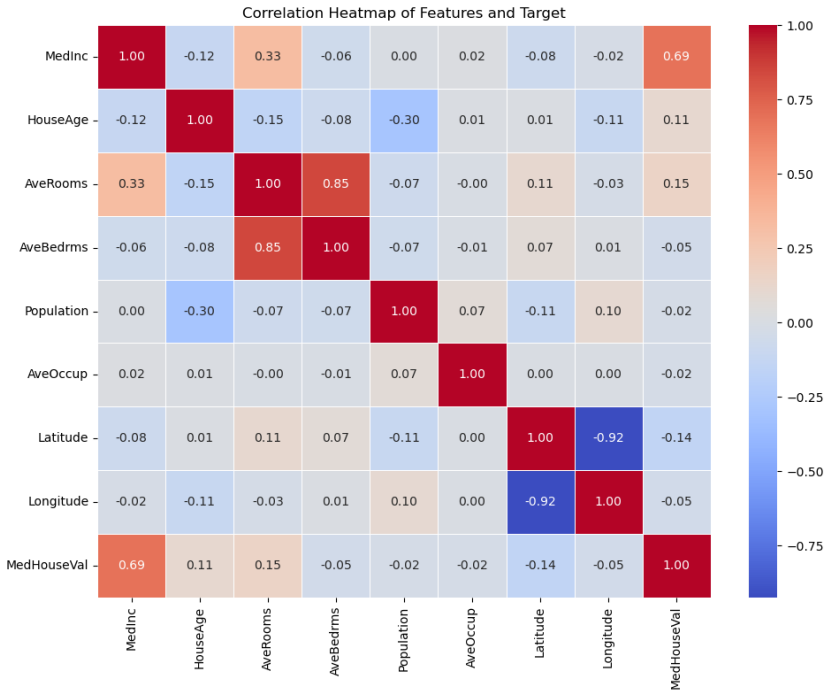

Figure 3: Correlation Heatmap of Features and Target

4.4 Discussion and Insights

The experimental results provide several valuable insights:

1. **Income as a Key Predictor:** The strong positive correlation between **MedInc** and housing prices aligns with expectations, highlighting income levels as a primary determinant of purchasing power and demand.
2. **Geographic Influence:** The negative correlations with **Latitude** and **Longitude** suggest that housing prices vary significantly based on location, likely due to regional economic and infrastructure factors.
3. **Room Usage Efficiency:** The negative coefficient for **AveRooms** may indicate inefficiencies in space utilization in larger homes, which could be less desirable in the California housing market.
4. **Potential Model Limitations:** Slight dispersion in high-price predictions indicates that the model might benefit from including additional features or nonlinear terms to capture complex interactions.

4.5 Recommendations for Future Work

To further enhance the model, the following recommendations are proposed:

- **Feature Engineering:** Incorporate additional features such as proximity to amenities (e.g., schools, parks, or transportation) and environmental factors (e.g., pollution levels or climate conditions).
- **Nonlinear Modeling:** Explore polynomial regression or machine learning methods (e.g., random forests or gradient boosting) to capture potential nonlinear relationships.
- **Geospatial Analysis:** Conduct a detailed geospatial analysis to better understand the impact of location on housing prices.

5 Conclusion and Future Work

5.1 Conclusion

This project demonstrated the application of QR decomposition in solving multivariate linear regression problems. Using the California Housing Dataset, the QR decomposition-based regression model was implemented and evaluated. The main contributions of this work are as follows:

- QR decomposition provided a numerically stable and efficient solution to the regression problem, avoiding the need for matrix inversion.
- The model achieved a reasonable Mean Squared Error (MSE) of 0.2834, indicating strong predictive performance.
- Regression coefficients revealed key insights into feature importance, with Median Income being the most significant positive predictor of housing prices, and geographical features (Latitude and Longitude) having a negative impact.

5.2 Limitations

Despite its success, this study has some limitations:

- **Non-linear Relationships:** The linear regression model struggled to capture non-linear relationships between features and target variables.
- **Multicollinearity:** While QR decomposition improves numerical stability, multicollinearity among features may still affect coefficient estimates.
- **Outliers and Noise:** The presence of outliers or noise in the dataset could influence the model's performance and reduce its robustness.

5.3 Future Work

To address these limitations and further enhance the model, the following directions are proposed:

- **Model Extensions:** Explore non-linear models such as polynomial regression, decision trees, or neural networks to better capture complex patterns in the data. Regularization methods like ridge regression or Lasso could also be incorporated to mitigate multicollinearity.
- **Feature Engineering:** Perform feature selection and engineering to include additional relevant variables, such as transportation accessibility and school quality, which may improve prediction accuracy.
- **Dataset Improvements:** Use larger, more diverse datasets that account for temporal and regional variations in housing prices. Removing outliers and addressing noise in the data could further improve model performance.
- **Algorithm Optimization:** Compare QR decomposition with alternative matrix factorization methods, such as Householder reflections or Givens rotations, for better performance in high-dimensional datasets. Additionally, integrating QR decomposition into distributed computing frameworks (e.g., Spark) could enable efficient processing of large-scale data.