



Università
di Catania

UNIVERSITY OF CATANIA

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

Alfio Spoto

Numerical Analysis of Linear Regression: Comparative Study
of Closed-Form and Iterative Method

FINAL PROJECT REPORT

Professor: Sebastiano Boscarino

Academic Year 2024 - 2025

Contents

1	Introduction	3
1.1	Background and Motivation	3
1.2	Study Objectives	3
1.3	Structure of the Document	4
2	Theoretical Background	5
2.1	Linear Regression Model	5
2.2	Cost Function	6
2.3	Closed-Form Solutions	6
2.3.1	Normal Equations (Least Squares)	6
2.3.2	Singular Value Decomposition (SVD)	7
2.3.3	QR Decomposition	7
2.4	Iterative Methods	8
2.4.1	Conjugate Gradient	8
2.4.2	Adam Optimization	8
2.5	Theoretical Comparison of Methods	10
2.6	The Problem of Ill-Conditioning and Multicollinearity	10
2.6.1	Definition and Causes	10
2.6.2	Consequences of Multicollinearity	11
2.6.3	Regularization Techniques	11
2.6.3.1	Ridge Regression (L2 Regularization)	11
2.6.3.2	Lasso Regression (L1 Regularization)	12
3	Practical Application and Experimental Results	13
3.1	Dataset and Preprocessing	13
3.1.1	Dataset Description	13
3.1.2	Exploratory Data Analysis	14
3.1.3	Data Preprocessing	15
3.2	Methods	15
3.3	Results and Analysis	15
3.3.1	Evaluation Metrics	15
3.3.2	Performance Comparison	16
3.3.3	Convergence Analysis of Iterative Methods	18
3.3.4	Inference Times	19
3.4	Linear vs. Polynomial Model	20
3.4.1	Performance Comparison	20
3.4.2	Interpretation of Results	20

<i>CONTENTS</i>	2
Conclusion	21
3.5 Conclusions and Final Considerations	21
3.5.1 Equivalence of Direct Methods (Closed-Form Solutions)	21
3.5.2 Efficiency of Iterative Methods	21
3.5.3 Importance of Exploratory Analysis	21
3.5.4 Practical Considerations	21
Bibliography	23

Chapter 1

Introduction

1.1 Background and Motivation

Linear regression represents one of the fundamental techniques in statistical analysis and machine learning, frequently constituting the initial paradigm for understanding complex relationships between variables [1]. Despite its apparent conceptual simplicity, the efficient and numerically stable implementation of linear regression algorithms presents non-trivial computational challenges, particularly in contexts characterized by high-dimensional datasets or ill-conditioned problems [2, 3]. This study explores and compares various numerical methods for solving linear regression problems, with particular emphasis on their mathematical foundations, computational properties, and empirical performance. The objective is to provide an in-depth analysis of the similarities and differences between closed-form approaches and iterative methods [4, 5], evaluating their behavior on a real dataset of global health and socioeconomic indicators [6].

1.2 Study Objectives

The main objectives of this work are:

- Analyze and compare the mathematical foundations of different methods for solving linear systems applied to regression, with particular attention to numerical stability and computational complexity [7, 2].
- Implement and empirically evaluate both direct methods (Least Squares, Singular Value Decomposition, QR Decomposition) and iterative methods (Conjugate Gradient [8], Adam optimization [9]) on a real dataset.
- Examine the convergence behavior of iterative methods and their performance characteristics in terms of accuracy, computational efficiency, and numerical stability [10, 11].
- Address practical issues such as ill-conditioning and multicollinearity that frequently emerge in the application of regression models to real data [12, 1].
- Extend the analysis to polynomial regression to capture non-linear relationships present in the dataset, comparing its performance with standard linear approaches[1].

1.3 Structure of the Document

The paper is organized according to the following structure: In Chapter 2, the theoretical background of linear regression is presented, and the five analyzed methods are described in detail: three closed-form solutions (Least Squares, SVD [2], QR [7]) and two iterative methods (Conjugate Gradient [8, 13], Adam [9]). Particular attention is dedicated to the mathematical properties that determine the computational efficiency and numerical stability of each approach [3, 2]. Chapter 3 is devoted to practical application, where the dataset used [6], the experimental results, and an in-depth analysis of the performance of various methods are described. The problem of multicollinearity [12] is also addressed, and linear and polynomial regression models are compared in terms of predictive capabilities and adequacy to the considered dataset [1]. Finally, in the Conclusions, the main results are summarized, practical implications are discussed, and directions for future research in computational linear algebra applied to regression problems are suggested. This study aims to provide both a rigorous theoretical understanding and practical insights for researchers and professionals working with regression models in various application domains, from computational statistics to machine learning [1], from econometrics to health informatics.

Chapter 2

Theoretical Background

2.1 Linear Regression Model

Linear regression is a statistical technique that models the relationship between a dependent (or target) variable y and one or more independent variables (or predictors) x through a linear function [1]. Given a collection of m training examples with n features, the linear regression model can be expressed as:

$$h_{\theta}(x_i) = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \cdots + \theta_n x_{in} \quad (2.1)$$

Where:

- $h_{\theta}(x_i)$ is the predicted value for the i -th example
- θ_j are the model parameters (weights)
- θ_0 is the intercept (or bias)
- x_{ij} is the value of the j -th feature for the i -th example

In matrix notation, we can express this more compactly as:

$$h_{\theta}(X) = \theta^T X \quad (2.2)$$

Where $X \in \mathbb{R}^{m \times (n+1)}$ is the design matrix with an additional column of ones for the intercept term:

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1n} \\ 1 & x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} \quad (2.3)$$

And $\theta \in \mathbb{R}^{(n+1) \times 1}$ is the parameter vector $[\theta_0, \theta_1, \dots, \theta_n]^T$.

2.2 Cost Function

The objective of linear regression is to find the optimal values of θ that minimize the error between the model's predictions and the actual values [5]. The most commonly used cost function is the Mean Squared Error (MSE):

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i)^2 \quad (2.4)$$

In matrix form, this can be expressed as:

$$J(\theta) = \frac{1}{m} (\theta^T X - y)^T (\theta^T X - y) \quad (2.5)$$

This cost function is convex and differentiable, so we can find the global minimum by setting its gradient to zero [5]:

$$\nabla_{\theta} J(\theta) = \frac{2}{m} X^T (\theta^T X - y) = 0 \quad (2.6)$$

This leads us to the normal equations:

$$X^T \theta^T X = X^T y \quad (2.7)$$

Various numerical methods can be used to solve this system of linear equations, as we will see in the following sections [2, 3].

2.3 Closed-Form Solutions

Closed-form solutions directly find the optimal value of the parameters through algebraic operations, without requiring an iterative optimization process [3].

2.3.1 Normal Equations (Least Squares)

The most direct approach to solving the normal equations is to calculate [1, 2]:

$$\theta = (X^T X)^{-1} X^T y \quad (2.8)$$

This formula assumes that the matrix $X^T X$ is invertible, which is true when the columns of X are linearly independent. In practice, directly computing the inverse is often avoided in favor of more numerically stable methods [7, 2].

2.3.2 Singular Value Decomposition (SVD)

Singular value decomposition (SVD) decomposes the design matrix X into the product of three matrices [2, 7]:

$$X = U\Sigma V^T \quad (2.9)$$

Where:

- $U \in \mathbb{R}^{m \times m}$ is an orthogonal matrix containing the left singular vectors
- $\Sigma \in \mathbb{R}^{m \times (n+1)}$ is a diagonal matrix containing the non-negative singular values in decreasing order
- $V \in \mathbb{R}^{(n+1) \times (n+1)}$ is an orthogonal matrix containing the right singular vectors

The solution using SVD is [2]:

$$\theta = V\Sigma^+U^T y \quad (2.10)$$

Where Σ^+ is the pseudoinverse of Σ , obtained by taking the reciprocal of each non-zero singular value and transposing the resulting matrix.

SVD is particularly useful when $X^T X$ is ill-conditioned or singular, as it allows identification and management of small singular values that might cause numerical instability [2, 3, 14].

2.3.3 QR Decomposition

QR decomposition expresses the design matrix X as the product of two matrices [7, 2]:

$$X = QR \quad (2.11)$$

Where:

- $Q \in \mathbb{R}^{m \times m}$ is an orthogonal matrix ($Q^T Q = I$)
- $R \in \mathbb{R}^{m \times (n+1)}$ is an upper triangular matrix

Substituting this decomposition into the normal equations, we obtain [7]:

$$X^T \theta^T X = X^T y \quad (2.12)$$

$$R^T Q^T Q R \theta = R^T Q^T y \quad (2.13)$$

$$R^T R \theta = R^T Q^T y \quad (\text{since } Q^T Q = I) \quad (2.14)$$

Assuming that R has full rank, we can further simplify:

$$\theta = R^{-1} Q^T y \quad (2.15)$$

In practice, instead of explicitly calculating R^{-1} , the triangular system is solved using back substitution, which is computationally more efficient and numerically more stable [2, 3].

2.4 Iterative Methods

Iterative methods progressively approximate the optimal solution through a series of parameter updates, starting from a random initialization and incrementally improving [4, 5].

2.4.1 Conjugate Gradient

The Conjugate Gradient method is an iterative algorithm for solving systems of linear equations with symmetric and positive definite matrices [8, 13]. For linear regression, we apply it to the normal equations $X^T \theta^T X = X^T y$.

The algorithm proceeds by generating a sequence of search directions $\{p_k\}$ that are conjugate with respect to $X^T X$, i.e., $p_i^T X^T X p_j = 0$ for $i \neq j$. This allows faster convergence than simple gradient descent, as it avoids "zigzagging" in the same directions [13, 11].

The main steps of the algorithm are:

Algorithm 1 Conjugate Gradient for Linear Regression [8, 13]

Input: Design matrix X , target vector y , tolerance ϵ

Output: Optimal parameters θ

```

1: Initialize  $\theta_0$  arbitrarily (typically to zero)
2: Calculate the initial residual  $r_0 = X^T y - X^T \theta^T X_0$ 
3: Set  $p_0 = r_0$ 
4: for  $k = 0, 1, 2, \dots$  until convergence do
5:    $\alpha_k = \frac{r_k^T r_k}{p_k^T X^T X p_k}$ 
6:    $\theta_{k+1} = \theta_k + \alpha_k p_k$ 
7:    $r_{k+1} = r_k - \alpha_k X^T X p_k$ 
8:   if  $\|r_{k+1}\| < \epsilon$  then
9:     return  $\theta_{k+1}$ 
10:  end if
11:   $\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$ 
12:   $p_{k+1} = r_{k+1} + \beta_k p_k$ 
13: end for
```

The beauty of the Conjugate Gradient is that, for a problem in dimension n , it theoretically converges in at most n iterations in exact arithmetic [8, 11]. In practice, rounding errors may require more iterations, but convergence is still very rapid for well-conditioned problems [13].

2.4.2 Adam Optimization

Adam (Adaptive Moment Estimation) is a gradient-based optimization algorithm that combines the advantages of two other extensions of stochastic gradient descent: AdaGrad

and RMSProp [9]. It is particularly popular in neural network training but can also be applied to linear regression [10].

Adam maintains estimates of the first moment (the mean) and the second moment (the uncentered variance) of the gradients, which are updated exponentially [9]:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2.16)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2.17)$$

Where $g_t = \nabla_{\theta} J(\theta_t)$ is the gradient of the cost function at step t , and $\beta_1, \beta_2 \in [0, 1)$ are decay rates.

Since m_t and v_t are initialized to zero, bias correction factors are applied [9]:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2.18)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2.19)$$

Finally, the parameters are updated as follows:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \quad (2.20)$$

Where η is the learning rate and ϵ is a small value to avoid division by zero.

Algorithm 2 Adam for Linear Regression [9, 10]

Input: Design matrix X , target vector y , learning rate η , decay rates β_1, β_2 , ϵ , number of epochs

Output: Optimal parameters θ

```

1: Initialize  $\theta$  arbitrarily
2: Initialize  $m_0 = 0, v_0 = 0, t = 0$ 
3: for epoch = 1, 2, ..., number of epochs do
4:    $t = t + 1$ 
5:    $g_t = \frac{2}{m} X^T (\theta^T X_{t-1} - y)$  ▷ Gradient of the MSE function
6:    $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$ 
7:    $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$ 
8:    $\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$ 
9:    $\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$ 
10:   $\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$ 
11: end for
12: return  $\theta_t$ 

```

Unlike the Conjugate Gradient, Adam does not guarantee convergence in a finite number of steps, but its ability to adapt the learning rate for each parameter makes it robust to different feature scales and hyperparameter configurations [9, 10].

2.5 Theoretical Comparison of Methods

From a theoretical perspective, there is a fundamental distinction between closed-form methods and iterative methods [4, 5]:

Table 2.1: Theoretical Comparison between Direct and Iterative Methods [7, 4, 5]

Characteristic	Direct Methods	Iterative Methods
Convergence	Exact in a single computation step	Approximate, improves with the number of iterations
Computational complexity	$O(n^3)$ for $n \times n$ matrices	$O(kn^2)$ where k is the number of iterations
Numerical stability	May suffer from problems in the presence of multicollinearity or ill-conditioning	Can be more robust with appropriate regularization or preconditioning techniques
Scalability	Inefficient for large datasets	More suitable for large-scale problems
Hyperparameters	None	Requires calibration (e.g., learning rate, tolerance)
Applicability	Specific to linear problems	Easily extendable to non-linear problems

2.6 The Problem of Ill-Conditioning and Multicollinearity

A crucial aspect in the application of linear regression methods, especially closed-form ones, is the problem of ill-conditioning and multicollinearity [12, 1].

2.6.1 Definition and Causes

Multicollinearity occurs when two or more predictor variables in the model are strongly correlated with each other. From a mathematical perspective, this means that some columns of the design matrix X are nearly linearly dependent, making the matrix $X^T X$ almost singular, with a determinant close to zero [2, 3].

The condition number of a matrix, defined as the ratio between its largest and smallest singular values, quantifies this problem [7, 2]:

$$\kappa(X^T X) = \frac{\sigma_{\max}(X^T X)}{\sigma_{\min}(X^T X)} = \frac{\sigma_{\max}^2(X)}{\sigma_{\min}^2(X)} \quad (2.21)$$

A high condition number indicates that the matrix is ill-conditioned, which can lead to:

- Amplification of rounding errors in numerical solutions
- Instability in parameter estimates
- High variance in estimated coefficients

In linear regression, multicollinearity can arise for various reasons [1]:

- Intrinsically correlated variables (e.g., height and weight)
- Redundancy in measurements (e.g., temperature in Celsius and Fahrenheit)
- Small sample size relative to the number of predictors
- Inclusion of interaction or polynomial terms correlated with the original variables

2.6.2 Consequences of Multicollinearity

The main consequences of multicollinearity include [12, 1]:

- **Unstable coefficients:** Small changes in the data can cause large variations in the estimated coefficients.
- **High standard errors:** Coefficients are estimated with very wide confidence intervals.
- **Misleading interpretation:** It becomes difficult to determine the relative importance of predictor variables.
- **Overfitting:** The model may excessively fit the training data, showing poor generalization capability.

2.6.3 Regularization Techniques

To address the problem of multicollinearity, regularization techniques add a penalty term to the cost function, limiting the magnitude of the coefficients [1]:

2.6.3.1 Ridge Regression (L2 Regularization)

Ridge regression modifies the MSE cost function by adding a penalty term proportional to the sum of the squared coefficients [12]:

$$J_{\text{Ridge}}(\theta) = \frac{1}{m} \|\theta^T X - y\|^2 + \lambda \|\theta\|_2^2 \quad (2.22)$$

Where $\|\theta\|_2^2 = \sum_{j=1}^n \theta_j^2$ is the squared L2 norm of the parameter vector (excluding the intercept) and $\lambda \geq 0$ is the regularization parameter.

The closed-form solution becomes [12, 1]:

$$\theta_{\text{Ridge}} = (X^T X + \lambda I)^{-1} X^T y \quad (2.23)$$

Where I is the identity matrix (with the element corresponding to the intercept set to zero).

Adding λI to $X^T X$ improves the conditioning of the matrix, stabilizing the solution. A larger value of λ leads to greater shrinkage of the coefficients towards zero, reducing the variance but potentially increasing the bias [1].

2.6.3.2 Lasso Regression (L1 Regularization)

Lasso Regression (Least Absolute Shrinkage and Selection Operator) uses the L1 norm instead of the L2 norm [15, 1]:

$$J_{\text{Lasso}}(\boldsymbol{\theta}) = \frac{1}{m} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|^2 + \lambda \|\boldsymbol{\theta}\|_1 \quad (2.24)$$

Where $\|\boldsymbol{\theta}\|_1 = \sum_{j=1}^n |\theta_j|$ is the L1 norm of the parameter vector.

Unlike Ridge Regression, Lasso does not have a closed-form solution and requires numerical optimization methods [15]. The distinctive feature of Lasso is its ability to induce sparsity, driving some coefficients exactly to zero and thus performing automatic feature selection [15, 1].

Chapter 3

Practical Application and Experimental Results

3.1 Dataset and Preprocessing

3.1.1 Dataset Description

In this study, the "Life Expectancy Data" dataset provided by the World Health Organization (WHO) [6] is used, which contains information on various health, socioeconomic, and demographic indicators for different countries during the period 2000-2015. The dataset comprises 2938 observations, each representing a country-year combination, with 22 variables.

The target variable is life expectancy at birth, expressed in years. The main predictor variables include:

Table 3.1: Main Variables in the Life Expectancy Dataset [6]

Category	Variable	Description
Demographic	Year	Year of observation
	Population	Population size
	Schooling	Mean years of schooling (adults 25+)
	Income_composition	Human Development Index in relation to income
Health Indicators	Adult Mortality	Adult mortality rate (per 1000 population)
	Infant deaths	Number of infant deaths
	Hepatitis B	Hepatitis B vaccination coverage (%)
	Measles	Number of reported measles cases
	BMI	Average Body Mass Index
	HIV/AIDS	Deaths due to HIV/AIDS (per 1000 live births)
Economic Factors	GDP	Gross Domestic Product per capita
	Percentage expenditure	Health expenditure as a percentage of GDP
	Total expenditure	Government expenditure on health (% of total budget)

3.1.2 Exploratory Data Analysis

Prior to applying regression methods, an exploratory analysis was conducted to understand the distribution of variables and their relationships [1].

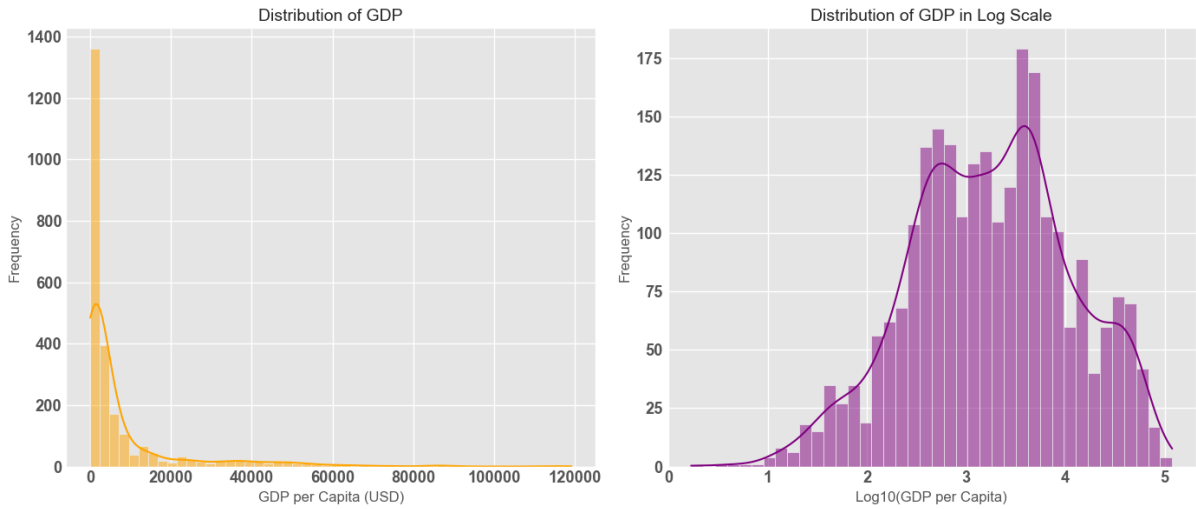


Figure 3.1: Distribution of GDP per Capita (Left: Raw Values, Right: Logarithmic Scale)

Figure 3.1 illustrates the distribution of GDP per capita in the dataset. In the left graph, a marked positive skewness is observed, with most countries having relatively low GDP values and few countries with significantly high values. When the same distribution is visualized on a logarithmic scale (right graph), GDP shows a distribution closer to normality, suggesting that a log-transformation might be appropriate when using GDP as a predictor in regression models [1].

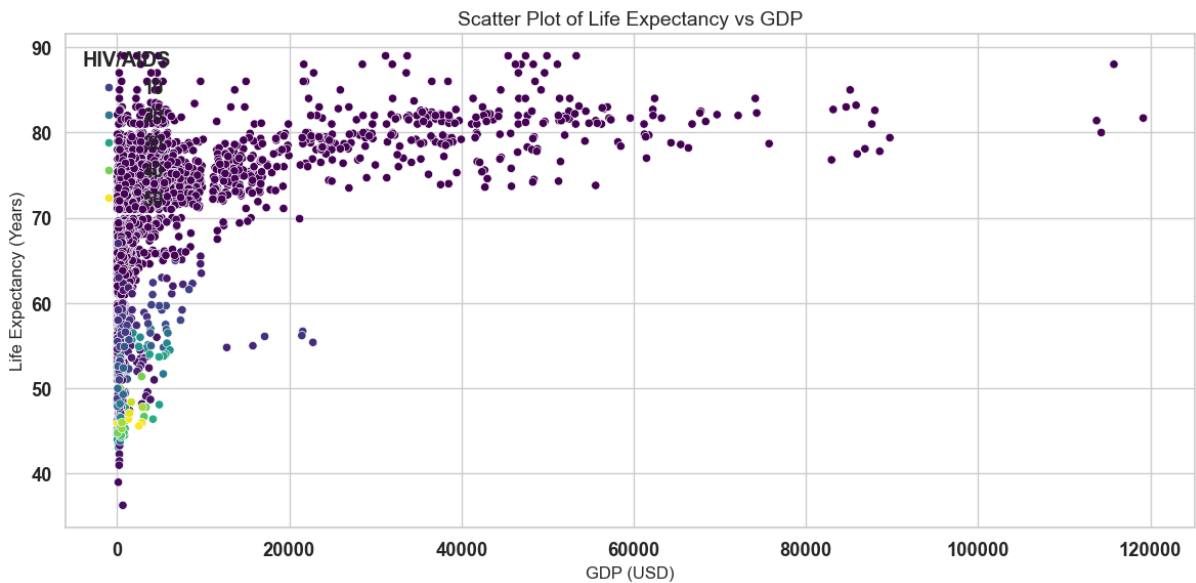


Figure 3.2: Scatter Plot of Life Expectancy vs GDP, Colored by HIV/AIDS Prevalence

Figure 3.2 visualizes the relationship between life expectancy and GDP per capita, with points colored according to HIV/AIDS prevalence. Several patterns are evident:

- A clear positive correlation between GDP and life expectancy is observed, following a logarithmic type relationship [1].
- The relationship is more pronounced at lower GDP levels but tends to plateau at higher levels, suggesting diminishing returns.
- Countries with high HIV/AIDS prevalence (indicated by warmer tones) tend to have lower life expectancy regardless of GDP.
- There is significant variability in life expectancy among countries with comparable GDP levels, indicating the significant influence of other factors.

This non-linear relationship between GDP and life expectancy suggests that a transformation of the GDP variable or a polynomial regression model might be more appropriate than a simple linear model [1].


3.1.3 Data Preprocessing

Before applying regression methods, the following preprocessing steps were implemented[1]:

- **Missing value imputation:** Missing values were imputed using the median for numerical variables, stratifying by country where possible.
- **Feature scaling:** All predictor variables were standardized by subtracting the mean and dividing by the standard deviation, to ensure comparable scales and improve the numerical stability of algorithms [2, 3].
- **Train-test split:** The dataset was divided into a training set (80%) and a test set (20%), preserving the temporal distribution of observations.

3.2 Methods

All five methods discussed in Chapter 2 were implemented using Python with the NumPy library for linear algebra operations. Below is a description of the implementation of each method.

For further implementation details, please consult the [GitHub Repository](#).  of the project.

3.3 Results and Analysis

In this section, the results of applying the five regression methods to the life expectancy dataset are presented, and an analysis of their performance is conducted.

3.3.1 Evaluation Metrics

To evaluate the models' performance, the following metrics were used [1]:

- **Mean Squared Error (MSE):** $\frac{1}{m} \sum_{i=1}^m (\hat{y}_i - y_i)^2$, which penalizes larger errors more heavily.
- **Mean Absolute Error (MAE):** $\frac{1}{m} \sum_{i=1}^m |\hat{y}_i - y_i|$, which is less sensitive to outliers.

- **Coefficient of determination (R^2):** $1 - \frac{\sum_{i=1}^m (\hat{y}_i - y_i)^2}{\sum_{i=1}^m (y_i - \bar{y})^2}$, which quantifies the proportion of variance explained by the model.
- **Inference time:** The computational time required to make a prediction after training.

3.3.2 Performance Comparison

The performance results of the five methods on the test set are reported in Table 3.2.

Table 3.2: Performance comparison of regression methods

Method	Test MSE	Train MSE	Test MAE	Train MAE	R^2	Inference Time (s)
LeastSquares	13.430957	13.058380	2.795885	2.784709	0.810892	0.000034
SVD	13.430979	13.058380	2.795890	2.784708	0.810891	0.000019
QR	13.430967	13.058380	2.795886	2.784709	0.810891	0.000015
ConjugateGradient	13.429089	13.058776	2.795196	2.784184	0.810918	0.000011
Adam	13.452832	13.062629	2.799948	2.787141	0.810583	0.000012

The results show extremely similar metrics (MSE, MAE, R^2) and parameters among the Least Squares, SVD, QR, and Conjugate Gradient methods, while Adam presents slight differences. This behavior can be justified by [2, 7, 8]:

1. **Linear nature of the problem:** The data follow a sufficiently linear relationship, leading all methods to converge towards the same theoretical solution. The small differences detected are attributable to numerical errors in the implementations [3].
2. **Parameter convergence:** The coefficients and intercept obtained with Least Squares, SVD, QR, and Conjugate Gradient are numerically identical, as evidenced by the following values:

Table 3.3: Model Coefficients and Intercept

Parameter	Value
θ_0 (Intercept)	69.1476
θ_1	-0.5426
θ_2	-2.2143
θ_3	-0.7177
θ_4	0.6625
θ_5	-0.1368
θ_6	0.0763
θ_7	0.5515
θ_8	-0.3360
θ_9	0.2093
θ_{10}	0.2395
θ_{11}	0.4069
θ_{12}	-2.7783
θ_{13}	0.1936
θ_{14}	0.1420
θ_{15}	-0.2758
θ_{16}	-0.0335
θ_{17}	1.8965
θ_{18}	2.8509

The consistency of these coefficients across all direct methods and in the Conjugate Gradient indicates that all have precisely identified the optimal solution to the linear problem $\min_{\theta} \|y - \theta^T X\|^2$. The slight differences in Adam’s coefficients, although not significant enough to substantially compromise predictive performance, can be attributed to the stochastic nature of the algorithm and its iterative convergence process [9, 10].

3. **Algebraic equivalence:** Least Squares, SVD, and QR are direct methods for solving $\theta^T X = y$, while the Conjugate Gradient is an iterative method that, for well-conditioned quadratic problems, converges to the same exact solution [8, 11]. This algebraic equivalence is confirmed by the substantial identity of the obtained coefficients, up to the first four decimal places.

It is particularly significant to note how all methods converge towards identical coefficients despite their different mathematical formulations [3, 7]. This confirms the uniqueness of the solution for well-specified and well-conditioned linear regression problems. The precision of the Conjugate Gradient approach in reaching exactly the same solution as the direct methods, but with significant computational efficiency, makes it particularly advantageous for larger datasets [13, 4]. The obtained coefficients also provide interpretative information about the dataset: for example, the relatively high and positive coefficient for parameter 18 (2.8509), which corresponds to years of schooling, suggests a strong positive impact of education on life expectancy, while the negative coefficient for parameter 12 (-2.7783), correlated with HIV/AIDS, indicates the significant negative impact of this disease on the target variable [6].

3.3.3 Convergence Analysis of Iterative Methods

A particularly interesting aspect is the comparison of convergence behavior between the two iterative methods: Conjugate Gradient and Adam [8, 9].

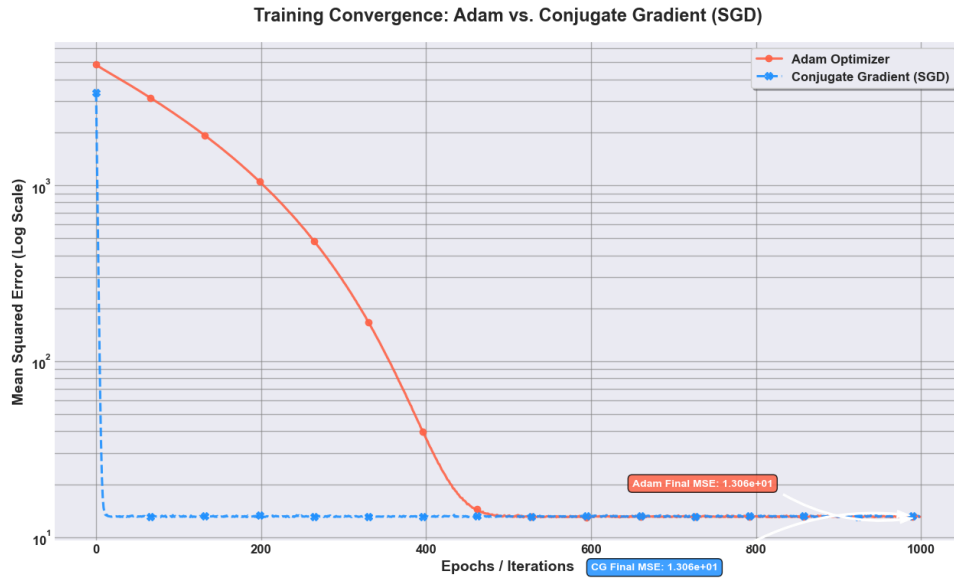


Figure 3.3: Convergence Behavior of Iterative Methods during Training (Logarithmic Scale)

Figure 3.3 shows the trend of the error (MSE) during iterations for both methods. It is observed that:

- **The Conjugate Gradient** demonstrates a drastically faster convergence compared to Adam. The error decreases almost vertically in the very first iterations (within 10 iterations), rapidly reaching a convergence plateau at about 13.06 MSE. This behavior is consistent with the theory, which guarantees convergence in a finite number of steps (at most equal to the dimensionality of the problem) for quadratic cost functions [8, 13, 11].
- **Adam** presents a significantly more gradual decline in error, requiring a considerably larger number of iterations (over 500) to approximate the same error level. This slower descent is characteristic of optimization algorithms based on stochastic gradient descent [9, 10].

Despite the markedly different convergence dynamics, both methods reach remarkably similar final error values:

- Training MSE for Adam: 13.062629
- Training MSE for Conjugate Gradient: 13.058776

This analysis highlights an important trade-off in the selection of the optimization method [10, 4]:

- **The Conjugate Gradient** proves extremely efficient for quadratic optimization problems such as linear regression, thanks to its ability to converge in very few iterations [8, 13]. However, it is specifically designed for quadratic cost functions.

- **Adam** requires more iterations but offers greater flexibility, being applicable to a wider range of non-convex optimization problems (such as neural networks) [9, 10]. Its slower convergence represents the trade-off for this versatility.

3.3.4 Inference Times

Inference times are negligible and similar for all methods (about 0.00002s) since they depend exclusively on calculating the product $\theta^T X$, which is common to all approaches. The minimal differences detected derive from implementation optimizations, with the ConjugateGradientRegressor being the most performant model (0.000011s) and the Least-SquareRegressor the least efficient (0.000034s).

It is important to note that, although training times may vary significantly between methods (with direct methods being more efficient for this moderately sized dataset), inference times are essentially equivalent since the prediction operation is identical for all models [10].

3.4 Linear vs. Polynomial Model

Based on the exploratory data analysis, particularly the non-linear relationship observed between GDP and life expectancy, linear and polynomial regression models were implemented and compared [1].

3.4.1 Performance Comparison

The comparison between the Linear Regression Model and the Polynomial Regression Model demonstrates that the latter provides superior fitting for the dataset, as evidenced by the evaluation metrics [1]:

Table 3.4: Comparison between Linear and Polynomial Models

Metric	Linear Model	Polynomial Model
MSE	13.43	8.11
R ²	0.81	0.88

- The **lower MSE** indicates that the polynomial model makes more accurate predictions with a lower error rate. The 40
- A **higher R²** implies that the polynomial model explains a greater proportion of the variance in the data. The polynomial model explains an additional 7

3.4.2 Interpretation of Results

The superior performance of the polynomial model confirms the initial observation from the exploratory data analysis: the relationships between the target variable (life expectancy) and the predictors (especially GDP) are not strictly linear [1].

The polynomial model allows capturing:

- **Diminishing returns:** The effect of GDP increase on life expectancy decreases at higher GDP levels, as observed in Figure 3.2.
- **Interaction effects:** The interaction terms in the polynomial model allow modeling how the effect of one variable may be conditioned by the level of another variable. For example, the impact of health expenditure on life expectancy might vary depending on the level of schooling [1].
- **Non-linear thresholds:** Some health indicators might present threshold effects that are more adequately modeled by polynomial terms.

However, it is important to emphasize that, although polynomial models offer greater flexibility, they also entail a greater risk of overfitting, especially with high-degree polynomials [1]. For this analysis, the polynomial degree was limited to 2 to ensure an adequate balance between model complexity and generalization capability.

Conclusions

3.5 Conclusions and Final Considerations

The comparative analysis of different methods for solving linear regression problems has produced several relevant insights for applied numerical analysis.

3.5.1 Equivalence of Direct Methods (Closed-Form Solutions)

The three direct methods (Least Squares, SVD, QR) have generated essentially identical results in terms of estimated parameters and performance metrics. This confirms their algebraic equivalence in the absence of significant numerical problems [2, 7]. The minimal differences observed are attributable to round-off errors in the numerical implementation [3]. However, it is important to note that this equivalence might not be preserved under conditions of strong multicollinearity or with ill-conditioned matrices [12]. In such scenarios, methods based on SVD and QR should offer greater numerical stability compared to the direct solution of the normal equations [2, 3].

3.5.2 Efficiency of Iterative Methods

The Conjugate Gradient demonstrated exceptionally rapid convergence, reaching the optimal solution in few iterations. This confirms its status as a reference algorithm for well-conditioned quadratic problems [8, 13]. Its efficiency makes it an excellent choice for large-scale problems where direct methods become impractical [4, 11]. Adam, on the other hand, showed slower convergence but still reached a solution of similar quality. Its flexibility and adaptability make it advantageous for more complex problems, especially those that are non-convex or have sparse gradients [9, 10].

3.5.3 Importance of Exploratory Analysis

Exploratory data analysis proved essential for identifying the non-linear nature of the relationship between GDP and life expectancy [1]. This observation motivated the implementation of the polynomial model, which significantly improved predictive performance. This underscores the importance of thorough preliminary analysis before applying any regression model, to identify potential non-linearities and appropriate variable transformations [1].

3.5.4 Practical Considerations

In practice, the choice of regression method should consider not only accuracy, but also:

- **Problem size:** For large datasets, iterative methods or optimized implementations like QR might be necessary [4, 10].
- **Numerical conditioning:** In the presence of multicollinearity, regularization techniques such as Ridge [12] or SVD with truncation of singular values [2] can improve stability.
- **Interpretability:** More complex models like polynomials can offer better performance but at the expense of interpretability [1].
- **Computational requirements:** If training time is critical, direct methods or the Conjugate Gradient might be preferable to Adam [5, 10].

In the context of life expectancy analysis, the polynomial model proved superior, suggesting that the relationships between socioeconomic factors, health indicators, and life expectancy are intrinsically non-linear [6, 1] and require more flexible models to be adequately captured.

Bibliography

- [1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.
- [2] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 4th edition, 2013.
- [3] James W. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997.
- [4] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2nd edition, 2003.
- [5] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.
- [6] World Health Organization. Life expectancy data, 2018.
- [7] Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*. SIAM, 1997.
- [8] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.
- [9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [10] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.
- [11] Jörg Liesen and Zdeněk Strakos. *Krylov Subspace Methods: Principles and Analysis*. Oxford University Press, 2012.
- [12] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
- [13] Jonathan Richard Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. School of Computer Science Technical Report CMU-CS-94-125, Carnegie Mellon University, 1994.
- [14] Richard B. Lehoucq, Danny C. Sorensen, and Chao Yang. *ARPACK Users’ Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. SIAM, 1998.
- [15] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.