Journal of Machine Learning Research 23 (2022) 1-18 Submitted 1/21; Revised 5/22; Published 9/22

# Coordinated Minima Search: An Efficient Approach for Optimizing Linear and Non-Linear Regression Models

**Rifat Hassan** H.RIFAT1609@GMAIL.COM
*Department of Electrical and Electronic Engineering*
*Bangladesh University of Engineering and Technology*
*Dhaka-1205, Bangladesh*

**Editor:** My editor

## Abstract

In this paper, we introduce a novel regression optimization technique called Coordinated Minima Search (CMS), designed to iteratively minimize the loss function by coordinating over each weight and bias variable. Unlike traditional gradient-based approaches, CMS simplifies the complex optimization of linear and non-linear regression models by treating each weight variable's loss function as a parabolic equation, iterating to discover the global minima for each. This method allows for both linear and non-linear relationships between features and target variables. Therefore it enhances the ability to model complex datasets. Through considering the squared error loss function a parabolic function of each weight variable, CMS efficiently identifies the best-fitting model while minimizing squared error. The results demonstrate that CMS provides a flexible and precise alternative to traditional regression techniques, in datasets with both linear and non-linear dependencies.

**Keywords:** coordinated minima search, non-linear regression, feature transformation, optimization, squared error minimization, regression model fitting

## 1 Introduction

Regression analysis is one of the foundational tools for modeling relationships between dependent and independent variables in statistical and machine learning applications. Traditional approaches to regression, such as Ordinary Least Squares (OLS) and gradient-based methods like Stochastic Gradient Descent (SGD), focus on minimizing a loss function to optimize the model's weights. These techniques are highly effective for linear models but often struggle with non-linear relationships, resulting in underfitting or suboptimal performance.

In this paper, we propose a novel approach, Coordinated Minima Search (CMS), which is designed to overcome the limitations of traditional regression by allowing both linear and non-linear feature transformations while iterating over each weight variable's loss function individually. The CMS approach provides a flexible and scalable solution for both small and moderately sized datasets, yielding superior model fits for complex data.

## 2 Related Work

### 2.1 Introduction to Regression Analysis and Optimization

Regression analysis is one of the most fundamental techniques used for modeling relationships between dependent and independent variables. An in-depth look at linear and non-linear regression has been provided by Trevor Hastie, Robert Tibshirani, & Jerome Friedman (2009), in their book The Elements of Statistical Learning. Various optimization techniques for machine learning applications have been discussed here.

### 2.2 Gradient-Based Optimization Techniques

In both linear and non-linear regression, two widely used optimization methods are Gradient Descent and Stochastic Gradient Descent (SGD). These gradient-based optimization methods are used to iteratively minimize error functions. Ian Goodfellow, Yoshua Bengio, & Aaron Courville (2016) in their book Deep Learning outlined the key techniques of gradient descent and its variants. Gradient-based optimization methods calculate the gradient of the loss function with respect to each weight and adjust the weights in the direction of the steepest descent. Though they are effective they can suffer from problems such as slow convergence, sensitivity to initial values and getting stuck in local minima, particularly in complex, non-convex problems.

### 2.3 Coordinate Descent Methods

An alternative to gradient-based methods is Coordinate Descent (CD). In this method optimization is performed by adjusting one parameter at a time while keeping others fixed. CD methods are known for their simplicity and efficiency, especially in problems with sparse data or high-dimensional spaces. However, they may be slower than gradient-based methods in some settings and can also get stuck in suboptimal solutions. Stephen J. Wright (2015) in the paper Coordinate Descent Algorithms explored coordinate descent methods and their applications in machine learning and optimization problems. Yu. Nesterov (2012) in his paper explored the efficiency of Coordinate Descent Methods on huge-scale optimization problems. Stephen Boyd & Lieven Vandenberghe (2004) in the book Convex Optimization provided a comprehensive overview of optimization techniques, including comparisons between coordinate descent, gradient descent, and quadratic optimization methods.

### 2.4 Non-Linear Regression and Specialized Optimization Techniques

In cases where the relationship between variables cannot be expressed as a simple linear function, non-linear regression techniques are used. Methods like Levenberg-Marquardt and Trust-Region Methods are often used in these cases. These methods are adaptations of gradient descent tailored for non-linear problems. They offer a balance between the speed of gradient descent and the accuracy of second-order methods. Jorge Nocedal & Stephen J. Wright (2006) in the book Numerical Optimization covered various optimization techniques, including methods for non-linear problems. Theoretical backgrounds for methods like Levenberg-Marquardt have been provided in it.

### 2.5 Coordinated Minima Search and Parabolic Minimization

The Coordinated Minima Search (CMS) method builds upon the idea of parabolic minimization. By focusing on one parameter at a time and breaking the squared error loss function into a parabolic form, CMS seeks to find the precise values of the weight variables that result in minimum error using the formula $x = \frac{-b}{2a}$, where a,b are derived from the quadratic form of the error function. This approach makes CMS particularly efficient for problems where traditional gradient methods struggle with convergence or accuracy. Compared to gradient-based methods, CMS avoids the pitfalls of slow convergence and local minima in non-convex problems. Additionally, CMS's reliance on parabolic minimization offers a clear advantage over traditional coordinate descent methods in terms of precision. By calculating the minimum value for each weight in the form of a quadratic equation, CMS reduces the likelihood of getting trapped in suboptimal solutions.

CMS also extends beyond linear models, allowing for polynomial, trigonometric, logarithmic, and even exponential transformations of the feature variables. Thus it accommodates a wide range of non-linear regression models. The ability to handle such diverse transformations makes CMS a flexible and powerful tool for regression problems that can be applied to various domains like finance, healthcare, engineering, etc.

### 2.6 Gaps in the Literature and Contribution of CMS

Despite the extensive research in optimization and regression methods, there is a gap in methods that can effectively handle both linear and non-linear models while avoiding common pitfalls like slow convergence or local minima. The Coordinated Minima Search method addresses these gaps by:

1. Providing a precise parabolic minimization for each weight variable, ensuring accurate convergence.

2. Allowing for flexible transformations of the feature variables, enabling both linear and non-linear regression models to be built.

3. Avoiding the computational inefficiency of gradient-based methods in high-dimensional spaces.

## 3 Methodology

### 3.1 Coordinated Minima Search Overview

The CMS method is designed to minimize the squared error loss function through coordinated minimization of the weight and bias terms. For a given regression model, the loss function is treated as a parabolic equation for each individual weight variable while holding the others constant. By iterating over each variable and applying backward iteration to update the weight variables, CMS efficiently converges to the optimal values that minimize the loss.

Let us have a data set with $n$ independent variables $(x_1, x_2, x_3, \ldots, x_n)$ and a dependent variable $y$. We want to find the weight variables $(w_1, w_2, w_3, \ldots, w_n)$ and the bias variable $b$

that will build the best-fit relationship between the dependent variable $y$ and independent variables $(x_1, x_2, x_3, \ldots, x_n)$.

If we consider only a linear relationship, it will be like this:

$$y = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n + b$$

The Squared Error Loss Function will be:

$$J = \sum (w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n + b - y)^2$$

We know that,

$$\begin{aligned}(a + b + c + \cdots + m + n)^2 = {} & (a^2 + b^2 + c^2 + \cdots + m^2 + n^2) \\ & + 2a(b + c + \cdots + m + n) \\ & + 2b(c + \cdots + m + n) + \cdots + 2m \cdot n\end{aligned}$$

Therefore, the Loss Function $J$ becomes

$$\begin{aligned}J &= \sum (w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n + b - y)^2 \\ &= \sum \big\{ (w_1^2x_1^2 + w_2^2x_2^2 + \cdots + w_n^2x_n^2 + b^2 + y^2) \\ &\quad + 2w_1x_1(w_2x_2 + \cdots + w_nx_n + b - y) \\ &\quad + 2w_2x_2(w_3x_3 + \cdots + w_nx_n + b - y) + \cdots + 2w_nx_n(b - y) + 2b(-y) \big\} \\ &= (w_1^2 \sum x_1^2 + w_2^2 \sum x_2^2 + \cdots + w_n^2 \sum x_n^2 + Nb^2 + \sum y^2) \\ &\quad + (2w_1w_2 \sum x_1x_2 + \cdots + 2w_1w_n \sum x_1x_n + 2w_1b \sum x_1 - 2w_1 \sum x_1y) \\ &\quad + (2w_2w_3 \sum x_2x_3 + \cdots + 2w_2w_n \sum x_2x_n + 2w_2b \sum x_2 - 2w_2 \sum x_2y) \\ &\quad + \cdots + (2w_nb \sum x_n - 2w_n \sum x_ny) - 2b \sum y \\ &= (\sum x_1^2)w_1^2 + (2w_2 \sum x_1x_2 + \cdots + 2w_n \sum x_1x_n + 2b \sum x_1 - 2 \sum x_1y)w_1 \\ &\quad + w_2^2 \sum x_2^2 + \cdots + w_n^2 \sum x_n^2 + Nb^2 + \sum y^2 \\ &\quad + 2w_2w_3 \sum x_2x_3 + \cdots + 2w_2w_n \sum x_2x_n + 2w_2b \sum x_2 - 2w_2 \sum x_2y \\ &\quad + \cdots + 2w_nb \sum x_n - 2w_n \sum x_ny - 2b \sum y.\end{aligned}$$

Now, if we consider the whole loss function $J$ as a function of $w_1$ only, it becomes something like this:

$$J = a \cdot w_1^2 + b \cdot w_1 + c$$

where

$$a = \sum x_1^2$$

$$b = 2w_2 \sum x_1x_2 + \cdots + 2w_n \sum x_1x_n + 2b \sum x_1 - 2 \sum x_1y$$

$$\begin{aligned} c = w_2^2 \sum x_2^2 + \cdots + w_n^2 \sum x_n^2 + Nb^2 + \sum y^2 \\ + 2w_2w_3 \sum x_2x_3 + 2w_2w_n \sum x_2x_n + \cdots \\ + 2w_2b \sum x_2 - 2w_2 \sum x_2y + \cdots \\ + 2w_nb \sum x_n - 2w_n \sum x_ny - 2b \sum y. \end{aligned}$$

This is analogous to the parabolic equation $y = ax^2 + bx + c$. Let us break down the equation $y = ax^2 + bx + c$:

$$y = ax^2 + bx + c$$

$$\Rightarrow y = a\left(x^2 + \frac{b}{a}x\right) + c$$

$$\Rightarrow y = a\left(x^2 + 2\frac{b}{2a}x + \left(\frac{b}{2a}\right)^2 - \left(\frac{b}{2a}\right)^2\right) + c$$

$$\Rightarrow y = a\left(x^2 + 2\frac{b}{2a}x + \left(\frac{b}{2a}\right)^2\right) - \frac{b^2}{4a} + c$$

$$\Rightarrow y - c + \frac{b^2}{4a} = a\left(x + \frac{b}{2a}\right)^2$$

$$\Rightarrow y - \left(c - \frac{b^2}{4a}\right) = a\left(x - \left(-\frac{b}{2a}\right)\right)^2$$

$$\Rightarrow \frac{y - \left(c - \frac{b^2}{4a}\right)}{a} = \left(x - \left(-\frac{b}{2a}\right)\right)^2$$

$$\Rightarrow \left(x - \left(-\frac{b}{2a}\right)\right)^2 = 4\left(\frac{1}{4a}\right)\left(y - \left(c - \frac{b^2}{4a}\right)\right)$$

We find that the minimum value of $y$ is $c - \frac{b^2}{4a}$, and it appears when the value of $x$ is $-\frac{b}{2a}$.

Here, this minimum value of $J$ appears when the value of $w_1$ is:

$$\begin{aligned} w_1 &= -\frac{b}{2a} \\ &= -\frac{2w_2 \sum x_1x_2 + \cdots + 2w_n \sum x_1x_n + 2b \sum x_1 - 2 \sum x_1y}{2\sum x_1^2} \\ &= f(w_2, \ldots, w_n, b) \end{aligned}$$

That is, this is a function of the weight variables $w_2$ to $w_n$ (except $w_1$), and the bias variable $b$.

And the minimum value of the loss function $J$ is another function of the weight variables $w_2$ to $w_n$ (except $w_1$), and the bias variable $b$. It will be:

$$J = g(w_2, \ldots, w_n, b)$$

$$= c - \frac{b^2}{4a}$$

$$\begin{aligned}
= &\left(w_2^2 \sum x_2^2 + \cdots + w_n^2 \sum x_n^2 + Nb^2 + \sum y^2\right) \\
&+ 2w_2w_3 \sum x_2x_3 + \cdots + 2w_2w_n \sum x_2x_n \\
&+ 2w_2b \sum x_2 - 2w_2 \sum x_2y + \dots \\
&+ 2w_nb \sum x_n - 2w_n \sum x_ny - 2b \sum y \\
&- \frac{\left(2w_2 \sum x_1x_2 + \cdots + 2w_n \sum x_1x_n + 2b \sum x_1 - 2 \sum x_1y\right)^2}{4 \sum x_1^2}.
\end{aligned}$$

$$= a_2 \cdot w_2^2 + b_2 \cdot w_2 + c_2$$

This is the most interesting part of the algorithm. Starting from the weight variable $w_1$, for each weight variable $w$, the minimum value of the loss function $J$ follows the parabolic equation

$$y = ax^2 + bx + c,$$

where $a$, $b$, and $c$ are functions of the next weight variables and the bias variable.

For each case, the minimum value of the loss function and the value of the weight variables for which it occurs can be obtained by the equations

$$y = c - \frac{b^2}{4a} \quad \text{and} \quad x = \frac{-b}{2a},$$

where they are functions of the next weight variables and the bias variable.

For example, in the case of $w_2$, both the minimum value of $J$ and the value of $w_2$ for which it occurs are functions of $w_3$ to $w_n$ and $b$, i.e., $f(w_2, \dots, w_n, b)$. For $w_3$, this is $f(w_4, \dots, w_n, b)$. Thus, for $w_n$, this is $f(b)$.

For the bias variable $b$, these are two single values. That means if we start from $w_1$ and follow the equations $y = c - \frac{b^2}{4a}$ and $x = \frac{-b}{2a}$ for each weight variable $w$, we will end up getting a single minimum value of $J$, along with a single value of $b$ for which it occurs. This is the minimum value of the loss function $J$ we seek.

Along the way, we have to preserve the functions of the $w$'s that result in the minimum loss function for future use.

Our primary target is to find the values of the weight variables $w$'s and the bias variable $b$ that result in the minimum loss function. We have already found the required value of $b$.

Now, remember, the value of $w_n$ that results in the minimum loss function was a function of only $b$, $f(b)$. We preserved it earlier. Now that we have the value of $b$, we can easily find the required value of $w_n$ by substituting the value of $b$ into the previously found function.

Similarly, the value of $w_{n-1}$ that results in the minimum value of $J$ was a function of $w_n$ and $b$, $f(w_n, b)$. Now that we have the values of $w_n$ and $b$, we can easily find it.

Thus, we can find all the required values of the $w$'s up to $w_1$ through a backward iteration. This process will give us the precise values of all the weight variables $w$'s and the bias variable $b$ that result in the minimum value of the loss function.

### 3.2 Non-Linear Transformations

Not only this technique will give us the precise values of $w$'s and $b$, but also the technique can help us build a non-linear regression model. In fact, this technique can provide us with the minimum value of the loss function for each model possible, of both linear and non-linear, and we can find out which model is the best.

The squared error loss function that we took was:

$$\begin{aligned} = \sum \Big\{ & \left(w_1^2 x_1^2 + w_2^2 x_2^2 + \cdots + w_n^2 x_n^2 + b^2 + y^2\right) \\ & + 2w_1 x_1 \left(w_2 x_2 + \cdots + w_n x_n + b - y\right) \\ & + 2w_2 x_2 \left(w_3 x_3 + \cdots + w_n x_n + \cdots + b - y\right) \\ & + \ldots \\ & + 2w_n x_n (b - y) + 2b(-y) \Big\}. \end{aligned}$$

Now, in place of directly putting the feature columns $x_1$, $x_2$, …, $x_n$ in the model, if we want to put any of their functions, like $f_1(x_1)$, $f_2(x_2)$, or so on, we can easily put them in the loss function equation, like this:

$$\begin{aligned} = \sum \Big\{ & \left(w_1^2 (f_1(x_1))^2 + w_2^2 (f_2(x_2))^2 + \cdots + w_n^2 (f_n(x_n))^2 + b^2 + y^2\right) \\ & + 2w_1 (f_1(x_1)) \left(w_2 (f_2(x_2)) + \cdots + w_n (f_n(x_n)) + b - y\right) \\ & + 2w_2 (f_2(x_2)) \left(w_3 (f_3(x_3)) + \cdots + w_n (f_n(x_n)) + \cdots + b - y\right) \\ & + \ldots \\ & + 2w_n (f_n(x_n))(b - y) + 2b(-y) \Big\}. \end{aligned}$$

All other processes remain the same. We will end up getting the precise values of the $w$'s and $b$, which will result in the minimum value of the loss function.

Let us consider a problem. Think for a moment that we have a data set with $n$ independent variables from $x_1$ to $x_n$ and a dependent variable $y$. We want to find out the best equation that fits the model, which can be both linear and non-linear. For non-linear equations, we will consider up to the 2nd power.

Therefore, each independent variable can take two functions each: $x$ and $x^2$. As a result, there will be a total of $2^n$ combinations possible.

The following table will help to illustrate it in detail.

| | | | | |
|---|---|---|---|---|
| $x_1$ | $x_2$ | $x_3$ | … | $x_n$ |
| $x_1^2$ | $x_2^2$ | $x_3^2$ | … | $x_n^2$ |

There are 2 rows and $n$ columns in the table. Therefore, the total number of possible combinations will be $n$ 2's multiplied together, which is $2^n$.

Now we can write a code in any familiar language (most probably in Python or R) that will follow the above-mentioned technique, and find out the minimum value of the loss function for each of the $2^n$ combinations. Seeing the minimum values, we can decide which combination will fit the best. Normally, the best-fit model is the one where the minimum value of the squared error loss function is the lowest.

After choosing the best-fit combination, we will follow the mentioned techniques to find the values of the weight variables $w$ and the bias variable $b$.

Not necessary to say, that we can include any possible function in our way to choose the best combination possible. That can include algebraic functions like $x$, $x^2$, trigonometric functions like $\sin(x)$, $\cos(x)$, logarithmic functions like $\ln(x)$ or $\log(x)$, or even exponential function like $\exp(x)$. But it is suggested to keep the number of functions included in the procedure as small as possible, because, with the increase of the number of functions, the number of possible combinations increases in an exponential way $(x^n)$, where $x$ is the number of included functions and $n$ is the number of feature columns.

### 3.3 Algorithm Steps

**1. Initialization:** We consider the whole loss function $J$ as a function of $w_1$ only,

$$J(w_1) = a \cdot w_1^2 + b \cdot w_1 + c,$$

which is in the form $y = ax^2 + bx + c$, where:

$$a = \sum (f_1(x_1))^2$$

$$b = 2w_2 \sum (f_1(x_1) f_2(x_2)) + \cdots + 2w_n \sum (f_1(x_1) f_n(x_n)) + 2b \sum f_1(x_1) - 2 \sum (f_1(x_1) y)$$

$$\begin{aligned} c = {} & w_2^2 \sum (f_2(x_2))^2 + \cdots + w_n^2 \sum (f_n(x_n))^2 + Nb^2 + \sum y^2 \\ & + 2w_2 w_3 \sum (f_2(x_2) f_3(x_3)) + \cdots + 2w_2 w_n \sum (f_2(x_2) f_n(x_n)) \\ & + 2w_2 b \sum f_2(x_2) - 2w_2 \sum (f_2(x_2) y) + \ldots \\ & + 2w_n b \sum f_n(x_n) - 2w_n \sum (f_n(x_n) y) - 2b \sum y \end{aligned}$$

**2. Breaking down the Equation:** Next, we break down the equation to get the value of $w_1$ that results in the minimum value of $J$:

$$w_1 = \frac{-b}{2a}$$

$$= \frac{-\left(2w_2 \sum (f_1(x_1) f_2(x_2)) + \cdots + 2w_n \sum (f_1(x_1) f_n(x_n)) + 2b \sum f_1(x_1) - 2 \sum (f_1(x_1) y)\right)}{2 \sum (f_1(x_1))^2}$$

This simplifies to a function of all other weight variables $w_2, \ldots, w_n$ and the bias variable $b$, except $w_1$:

$$f(w_2, \ldots, w_n, b)$$

We preserve this equation for future use.

Similarly, we find out the minimum value of the loss function $J$ as another function of the weight variables $w_2$ to $w_n$ (except $w_1$), and the bias variable $b$:

$$g(w_2, \ldots, w_n, b)$$

$$= c - \frac{b^2}{4a}.$$

$$\begin{aligned}
= \Big( & w_2^2 \sum (f_2(x_2))^2 + \cdots + w_n^2 \sum (f_n(x_n))^2 + Nb^2 + \sum y^2 \\
& + 2w_2 w_3 \sum (f_2(x_2) f_3(x_3)) + \cdots + 2w_2 w_n \sum (f_2(x_2) f_n(x_n)) \\
& + 2w_2 b \sum f_2(x_2) - 2w_2 \sum (f_2(x_2) y) + \ldots \\
& + 2w_n b \sum f_n(x_n) - 2w_n \sum (f_n(x_n) y) - 2b \sum y \Big) \\
& - \frac{\left(2w_2 \sum (f_1(x_1) f_2(x_2)) + \cdots + 2w_n \sum (f_1(x_1) f_n(x_n)) + 2b \sum f_1(x_1) - 2 \sum (f_1(x_1) y)\right)^2}{4 \sum (f_1(x_1))^2}
\end{aligned}$$

$$= a_2 \cdot w_2^2 + b_2 \cdot w_2 + c_2$$

$$= J(w_2)$$

Again in the form of $y = ax^2 + bx + c$.

**3. Forward Iteration:** We continue this process forward to find out the equations for the minimum values of the loss functions as functions of decreasing numbers of weight variables $w$'s and the bias variable $b$. At the same time, we get the weight variables $w$'s as functions of the next weight variables and the bias variable, resulting in this minimum value of the loss function. We preserve these equations for future use.

**4. Ending up with the Bias Variable and the Minimum Value of the Loss Function:** We will end up getting the minimum value of the loss function and the bias variable, both in full numerical forms.

**5. Backward Iteration to Get the Weight Variables:** Now we will use this bias variable $b$ to get the value of the last weight variable $w_n$ that results in the minimum value of the loss function, which was preserved as a function of $b$. Similarly, the values of all other weight variables that result in the minimum loss function will be found out by backward iteration, using the previously stored equations where a weight variable $w$ was written as a function of the next weight variables and the bias variable.

**6. Selecting the Best Fit Equation:** A similar process will be carried out for all possible combinations of the independent variables for all possible functions (a total of $x^n$ combinations, where $x$ is the number of included functions and $n$ is the number of independent variables). The minimum values of all the loss functions will be noted, and the equation with the smallest value of loss function will be chosen as the best-fit equation.

## 4 Experimental Setup and Results

### 4.1 Dataset and Experimental Setup

To evaluate the performance of CMS, we took the initiative to conduct experiments with real-world datasets. A suitable real-world dataset was chosen to show each step's breakdown in detail. Though CMS can handle a large number of features at a time, especially with advanced languages like python, we chose a dataset with a small number of independent variables so that the analysis of each independent variable can be shown in full detail. The selected dataset has four independent variables to predict one dependent variable. The number of data points, however, is not very small. A total number of 9568 data points are there. The full dataset is available here.

The data points were collected from a Combined Cycle Power Plant over 6 years (2006-2011) when the plant was set to work with a full load. The four independent variables of the dataset are respectively Ambient Temperature (AT), Exhaust Volume (V), Ambient Pressure (AP), and Relative Humidity (RH). They are used to predict the values of a dependent variable Power Output (PE).

### 4.2 Model Formation with CMS

We are showing the linear regression model here. That means the combination of indices of the independent variables we are taking is [1, 1, 1, 1]. The equation of our best-fit curve is:

$$y = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b$$

One can take any combination of indices and conduct the experiment.

The squared error loss function is:

$$\begin{aligned}
J(w_1, w_2, w_3, w_4, b) = {} & w_1^2 \sum x_1^2 + w_2^2 \sum x_2^2 + w_3^2 \sum x_3^2 \\
& + w_4^2 \sum x_4^2 + Nb^2 + \sum y^2 \\
& + 2w_1w_2 \sum x_1x_2 + 2w_1w_3 \sum x_1x_3 \\
& + 2w_1w_4 \sum x_1x_4 + 2w_1b \sum x_1 - 2w_1 \sum x_1y \\
& + 2w_2w_3 \sum x_2x_3 + 2w_2w_4 \sum x_2x_4 \\
& + 2w_2b \sum x_2 - 2w_2 \sum x_2y + 2w_3w_4 \sum x_3x_4 \\
& + 2w_3b \sum x_3 - 2w_3 \sum x_3y + 2w_4b \sum x_4 \\
& - 2w_4 \sum x_4y - 2b \sum y.
\end{aligned}$$

Putting the corresponding values in place, we get:

$$
\begin{aligned}
J(w_1, w_2, w_3, w_4, b) = {} & 4226228.0792w_1^2 + 29762163.2391w_2^2 \\
& + 9823745224.5548w_3^2 + 53459782.0308w_4^2 \\
& + 9568b^2 + 1978076968.9819002 \\
& + 21951075.3908w_1w_2 + 380602167.3846w_1w_3 \\
& + 26438023.3854w_1w_4 + 376045.96w_1b \\
& - 168554686.0126w_1 + 1052377528.7872w_2w_3 \\
& + 75074091.27w_2w_4 + 1039195.86w_2b \\
& - 468564745.7976w_2 + 1421606173.7234w_3w_4 \\
& + 19389725.72w_3b - 8811018334.0838w_3 \\
& + 1402840.6w_4b - 639260166.9092w_4 \\
& - 8694728.82b.
\end{aligned}
$$

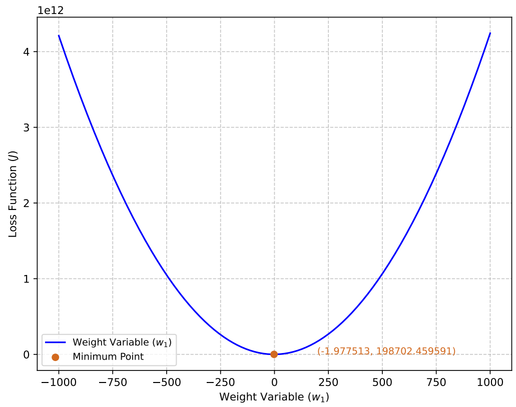
This is a function of $w_1, w_2, w_3, w_4$, and $b$. Considering it as a parabolic function of $w_1$ of the form $J = aw_1^2 + bw_1 + c$, the minimum value $c - \frac{b^2}{4a}$ is found as:

$$
\begin{aligned}
J(w_2, w_3, w_4, b) = {} & 1258631.142472323w_2^2 + 63951595.11838591w_2w_3 \\
& + 6414398.319104612w_2w_4 \\
& + 62602.42210770468w_2b - 30827294.008724272w_2 \\
& + 1254756098.601389w_3^2 + 231139306.66392922w_3w_4 \\
& + 2456907.0540330783w_3b \\
& - 1221238793.4384289w_3 + 12112683.185908273w_4^2 \\
& + 226624.8707311789w_4b - 112046224.94852197w_4 \\
& + 1202.9424722074054b^2 - 1195806.9059104733b \\
& + 297460019.0452943.
\end{aligned}
$$

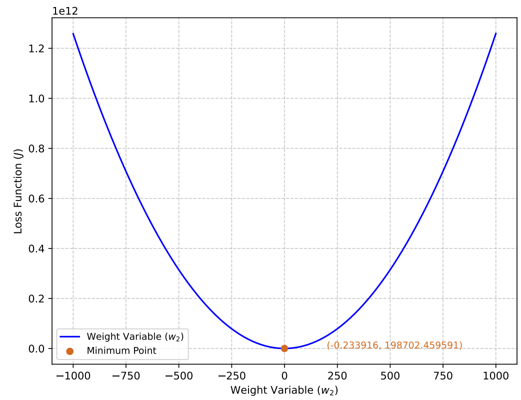
This is a function of $w_2, w_3, w_4$, and $b$. Considering it as a parabolic function of $w_2$, the minimum value is found as:

$$
\begin{aligned}
J(w_3, w_4, b) = {} & 442404016.212824w_3^2 + 68180124.10880119w_3w_4 + 866478.9226434017w_3b \\
& - 438064693.2136165w_3 \\
& + 3940212.235580419w_4^2 + 67103.60290732619w_4b \\
& - 33493209.613271415w_4 \\
& + 424.50486616401866b^2 - 429155.26078207954b \\
& + 108698986.55693269.
\end{aligned}
$$

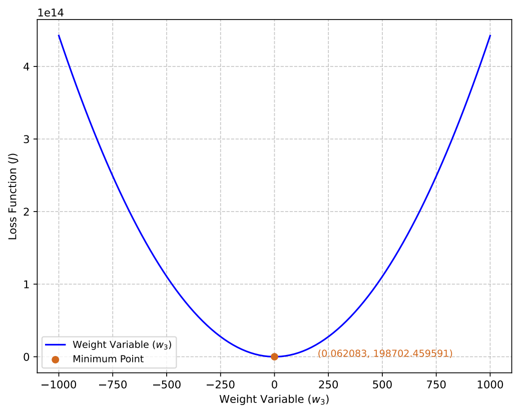
This is a function of $w_3, w_4$, and $b$. Considering it as a parabolic function of $w_3$, the minimum value is found as:



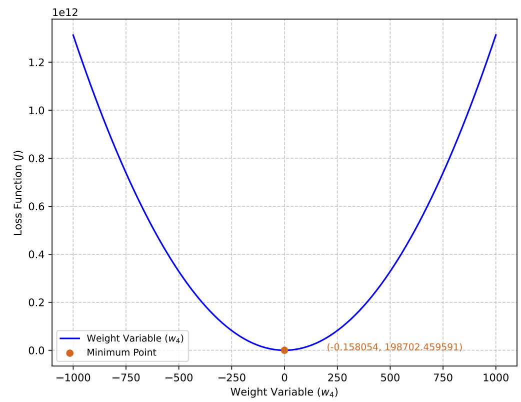

(a) Squared error loss function $J$ as a parabolic function of weight variable $w_1$.



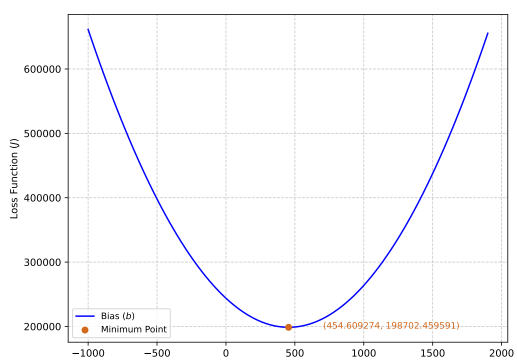

(b) Squared error loss function $J$ as a parabolic function of weight variable $w_2$.




(c) Squared error loss function $J$ as a parabolic function of weight variable $w_3$.




(d) Squared error loss function $J$ as a parabolic function of weight variable $w_4$.




(e) Squared error loss function $J$ as a parabolic function of bias variable $b$.

Figure 1: Main caption describing all figures.

$$J(w_4, b) = 1313354.6839215187w_4^2 + 335.85406407437404w_4b \\ + 262479.8204115853w_4 + 0.24011275358890316b^2 \\ - 165.23185651941458b + 257003.43305903673.$$

This is a function of $w_4$ and $b$. Considering it as a parabolic function of $w_4$, the minimum value is found as:

$$J(b) = 0.21864141120588582b^2 - 198.79282656884664b + 243888.99090438892.$$

This is a parabolic function of only $b$. The minimum value of this function is 198702.4595912306, and it appears when the value of $b$ is 454.6092743191532. This is shown in figure 1(e).

Therefore, the minimum value of the loss function is 198702.4595912306, and the value of the bias variable $b$ that results in this minimum value is 454.6092743191532. Putting $b = 454.6092743191532$ in $J(w_4, b)$, it becomes,

$$J(w_4) = 1313354.6839215187w_4^2 + 415162.19275757484w_4 + 231511.50356186475.$$

Again, the minimum value of this function is found to be 198702.4595912306. It appears when the value of $w_4$ is -0.1580541029167957. This is shown in figure 1(d).

Putting $w_4 = -0.1580541029167957$ and $b = 454.6092743191532$ in $J(w_3, w_4, b)$, it becomes,

$$J(w_3) = 442404016.212824w_3^2 - 54931487.33063036w_3 + 1903856.6793619245.$$

The minimum value of this function is found to be 198702.4595912306. It appears when the value of $w_3$ is 0.062082943777125296. This is shown in figure 1(c).

Putting $w_3 = 0.062082943777125296$, $w_4 = -0.1580541029167957$ and $b = 454.6092743191532$ in $J(w_2, w_3, w_4, b)$, it becomes,

$$J(w_2) = 1258631.142472323w_2^2 + 588828.9883958176w_2 + 267570.8448303938.$$

The minimum value of this function is found to be 198702.4595912306. It appears when the value of $w_2$ is -0.23391642258238726. This is shown in figure 1(b).

Putting $w_2 = 0.062082943777125296$, $w_3 = 0.062082943777125296$, $w_4 = -0.1580541029167957$ and $b = 454.6092743191532$ in $J(w_1, w_2, w_3, w_4, b)$, it becomes,

$$J(w_1) = 4226228.0792w_1^2 + 16714842.836514562w_1 + 16725612.851887941.$$

The minimum value of this function is found to be 198702.4595912306. It appears when the value of $w_1$ is -1.9775131066374658. This is shown in figure 1(a).

$$y = -1.9775131066374658x_1 - 0.23391642258238726x_2 + 0.062082943777125296x_3 \\ - 0.1580541029167957x_4 + 454.6092743191532$$

And the total squared error loss is: 198702.4595912306

Non-linear models were also formed from the dataset taking a maximum index of 2 for all the independent variables. As there were 4 independent variables, a total of $2^4 = 16$ models were possible. The models were numbered from 1 to 16 (Model 1 consists of the index combination [1, 1, 1, 1], Model 2 consists of the index combination [1, 1, 1, 2], etc. The last model consists of the index combination [2, 2, 2, 2].). The squared error losses of all the possible models have been shown in figure 2.

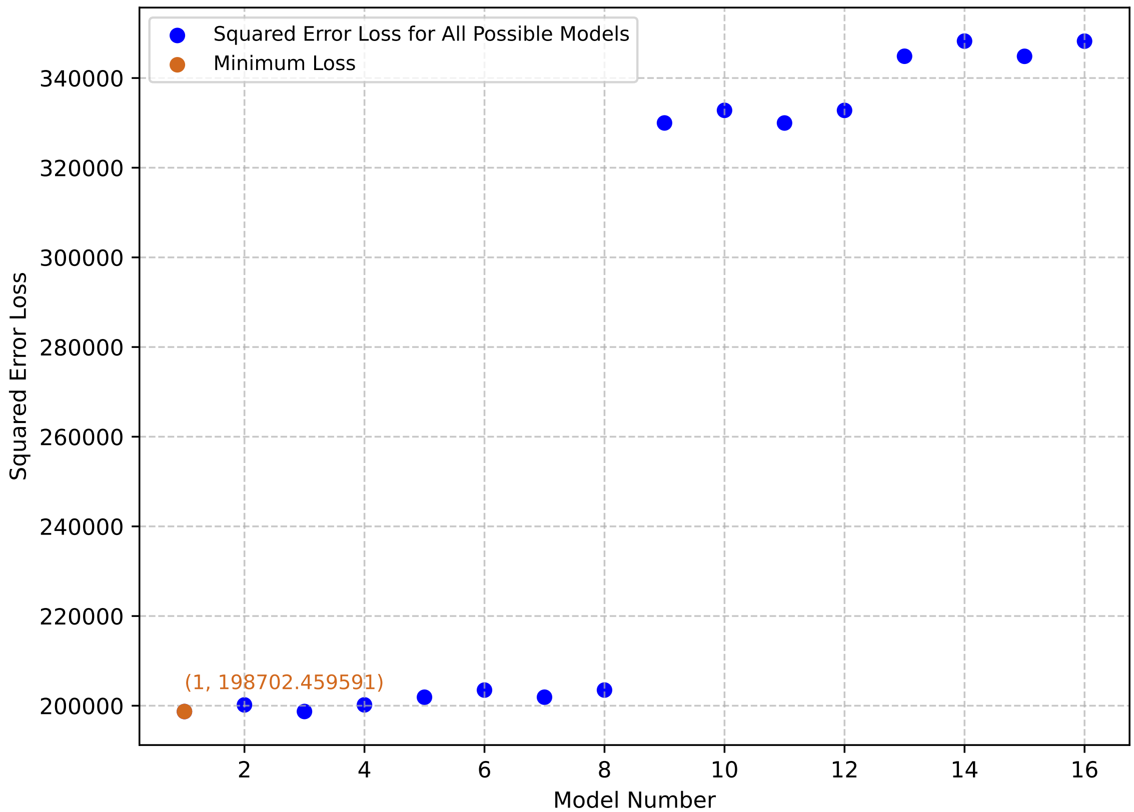



Figure 2: Your descriptive caption here.

### 4.3 Evaluation Metrics

The model formed using the Co-ordinated Minima Search (CMS) algorithm had the least squared error possible. It was further evaluated using:

- R-squared ($R^2$)
- Adjusted R-squared
- Residual Standard Error (RSE)
- F-statistic

### 4.4 Results

Evaluating the model with the already mentioned metrics, we got the following results:

- R-squared: 0.9286960898122537
- Adjusted R-squared: 0.9286513430848061
- RSE: 4.558793940891878
- F-statistic: 20754.503017004878

The results can be evaluated like this:

#### 4.4.1 Overall Model Fit

The R-squared value is found to be 0.9287. This suggests that the model appears to fit the data exceptionally well. 92.87% of the variance in the dependent variable is explained by the independent variables included in the model. The Adjusted R-squared is found to be 0.9287. This also indicates that the model complexity is justified by the explained variance.

#### 4.4.2 Residual Analysis

- **Residual Standard Error (RSE):** 4.5588

  The RSE indicates the average deviation of predicted values from the actual observed values. Here, the RSE is found to be 4.5588, whereas the range of the dependent variable $PE$ is 75.5 (420.26 to 495.76). Therefore, the RSE of 4.5588 represents about 6% of this range:

  $$\frac{4.5588}{75.5} \times 100 \approx 6\%.$$

  This indicates that the model's average prediction error is small relative to the scale of the dependent variable. This suggests it is a reasonably accurate model.

- **F-statistic:** 20754.50

  The F-statistic value is found here to be 20754.50. This very high F-statistic demonstrates that the predictors, as a group, are significantly related to the dependent variable. This confirms the model's overall statistical significance.

#### 4.4.3 Coefficient Estimates and Statistical Significance

Each predictor variable (Ambient Temperature, Exhaust Volume, Ambient Pressure, Relative Humidity) has p-values close to 0.000. This shows they have a statistically significant relationship with the dependent variable Power Output.

Also, the coefficients reflect each predictor's effect on the dependent variable, holding others constant. For instance:

- **Ambient Temperature (AT):** Coefficient is $-1.9775$. This suggests that with a unit increase in AT, the dependent variable decreases by approximately 1.98 units.

- **Exhaust Volume (V):** Coefficient is $-0.2339$. This indicates a reduction of about 0.23 units in the dependent variable per unit increase in V.
- **Ambient Pressure (AP):** Coefficient is 0.0621. This indicates an increase of about 0.06 units in the dependent variable per unit increase in AP.
- **Relative Humidity (RH):** Coefficient is $-0.1581$. This indicates a reduction of about 0.15 units in the dependent variable per unit increase in RH.
- **Power Output (PE):** Has a near-zero coefficient, suggesting minimal impact.

#### 4.4.4 Summary

Overall it is seen that the model performs quite well, explaining a substantial portion of the variance. A strong fit with an R-squared of 0.9287 explains 92.87% of the variance in the dependent variable. The Residual Standard Error (RSE) 4.5588, which is only 6% of the range of the dependent variable, indicates a low prediction error. A high F-statistic of 20754.50 confirms the model's statistical significance, with each predictor (AT, V, AP, RH) significantly contributing. The coefficients indicate each predictor's impact. Ambient Temperature has the largest effect, reducing the dependent variable by approximately 1.98 units per unit increase. Therefore it can be concluded that he regression model developed using the CMS approach, which minimizes the sum of squared errors, demonstrates strong functionality and reliability across all other evaluation metrics.

## 5 Installation in Python

The full Python code of the Co-ordinated Minima Search algorithm has been uploaded to Python Package Index (PyPI) with the name `cms_model`. One can comfortably install the package on their machine using the `pip` package manager. To install the package, the following command needs to be run in the terminal:

```
pip install cms_model
```

Once installed, the package can be imported and used within Python scripts or projects for regression analysis using the Co-ordinated Minima Search (CMS) method.

The full details of the package are available on https://pypi.org/project/cms-model/.

## 6 Discussion

### 6.1 Advantages of CMS

The primary advantage of the Co-ordinated Minima Search (CMS) method is that it provides the most precise output possible. Specifically, it identifies the weight variables and bias that result in the least possible squared error. Naturally, this yields the best regression model possible with the available data.

Unlike gradient descent or other similar optimization methods, there is no risk of becoming trapped in local minima. Another major advantage of CMS is its ability to handle both linear and non-linear models simultaneously. It does so by identifying the weight variables and biases that result in the least squared error in each case. For non-linear models, the CMS method computes the weight variables for each case while storing the corresponding value of the squared error loss function. Consequently, after evaluating all or a portion of the non-linear models, the best model can be selected easily based on the loss function value.

The major advantages of the CMS approach are summarized below:

- **Novelty of the Approach:** The Co-ordinated Minima Search (CMS) method offers a completely new and innovative way of tackling regression problems, particularly for both linear and non-linear models. This contributes valuable knowledge to the fields of machine learning, optimization, and statistical modeling.

- **Preciseness of the Algorithm:** CMS provides the most precise model possible by finding the weight variables and biases that result in the least squared error. Unlike Gradient Descent or similar optimization methods, CMS avoids getting trapped in local minima, making it one of the best approaches for regression problems in both linear and non-linear cases.

- **Robustness of the Method:** The CMS method can be considered a hyperparameter-free optimization technique, as it is not sensitive to hyperparameter choices. Unlike Gradient Descent, CMS does not depend on parameters like the learning rate, making the approach easy to compute. This enhances the likelihood of obtaining the best possible result with minimal effort and ensures consistent performance across different datasets.

- **Complexity Handling:** CMS minimizes loss functions by breaking them down into parabolic forms and systematically solving for each weight variable. This systematic approach may provide computational advantages and a clearer methodology for handling large-scale regression tasks involving complex data.

- **Application Scope:** The CMS technique is versatile enough to handle a variety of functions, including polynomial, trigonometric, logarithmic, and exponential models. In real-world applications, it opens new opportunities in domains like market analysis and finance, where analyzing large datasets with complex non-linear models is crucial.

### 6.2 Limitations

Despite its advantages, CMS has some limitations. It can become computationally expensive when applied to very large datasets. As this technique forms the first loss function taking all the available data into account, the availability of a very large number of data may surpass the machine's capacity, and throw overflow error. Also as it analyzes each feature column individually after forming the loss function, a very large number of feature columns may sometimes slow down the process. But nevertheless, all these depend on the capacity of the machine's processor, and while working with powerful processors, will hardly cause any issues.

### 6.3 Future Work

In spite of the limitations, Co-ordinated Minima Search brings a milestone in regular regression techniques by analyzing the loss function as a parabolic function of each feature column and finding out the weight variables that result in the least value of this loss function by fitting perfectly in the loss function. Though this paper is all about algebraic (polynomial) regression models, future works may extend it to all types of regression models like exponential, trigonometric, logarithmic, etc. Also in this paper, only the squared error loss function has been focused on, because this function acts as a parabolic function of weight variables whose minimum value can be easily found by the formula: $y_{\min} = c - \frac{b^2}{4a}$. However, there are scopes to extend this method to all types of loss functions, as the minimum value of any dependent function can be found by setting its first derivative with respect to the independent variable (s) to zero. Subsequently, the values of the independent variables that result in the loss function being minimum can be sorted out, along with the minimum value of the loss function. Thus this method can be extended to all types of loss functions by finding out the weight variables and bias value that result in the most perfect fitting of the model. These scopes are kept for future work.

## 7 Conclusion

Coordinated Minima Search offers a promising alternative to traditional regression techniques by introducing a coordinated approach to minimizing the loss function in both linear and non-linear regression models. Through the parabolic minimization of the loss function, CMS demonstrates significant improvements in predictive accuracy, especially in datasets with non-linear relationships. Given its potential applications across multiple domains and its capacity to model complex non-linear relationships, it represents a significant contribution to the field of regression analysis and optimization. The method's flexibility and precision make it a valuable tool for regression modeling, though future work is needed to extend its scalability.

## References

Tüfekci, S. (Year). *Prediction of full-load electrical power output of industrial gas turbine engines*. Available at https://www.semanticscholar.org/paper/Prediction-of-full-load-electrical-power-output-of-T%C3%BCfekci/f164ee353ada936ef97d0d379f8581336ddc9733.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.)*. Springer.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep learning*. Nature, 521(7553), 436–444. Available at https://arxiv.org/abs/1502.04759.

Boyd, S., & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press. Available at https://epubs.siam.org/doi/10.1137/100802001.