2022 2nd International Conference on Intelligent Technologies (CONIT)
Karnataka, India. June 24-26, 2022

Study of Lasso and Ridge Regression using ADMM

A.M. Abhishek Sai
Department of Computer Science and Engineering
Amrita Vishwa Vidyapeetham,
Amritapuri, India
amenu4aie20101@am.students.amrita.edu

Kalla Likhit Sai Eswar
Department of Computer Science and Engineering
Amrita Vishwa Vidyapeetham,
Amritapuri, India
amenu4aie20137@am.students.amrita.edu

Koganti Sri Sai Harshith
Department of Computer Science and Engineering
Amrita Vishwa Vidyapeetham,
Amritapuri, India
amenu4aie20139@am.students.amrita.edu

Perumalla Raghavendra
Department of Computer Science and Engineering
Amrita Vishwa Vidyapeetham,
Amritapuri, India
amenu4aie20156@am.students.amrita.edu

G.Yashwanth Kiran
Department of Computer Science and Engineering
Amrita Vishwa Vidyapeetham,
Amritapuri, India
amenu4aie20175@am.students.amrita.edu

Mithun V.
Department of Computer Science and Engineering
Amrita Vishwa Vidyapeetham,
Amritapuri, India
mithunv@am.amrita.edu

2022 International Conference on Intelligent Technologies (CONIT) | 978-1-6654-8407-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/CONIT55038.2022.9847706

***Abstract*—Convex optimization has several applications, including communication networks, data analytics, economics, and statistics. This technique emerged as the most efficient method of optimization. It can be performed using a variety of algorithms like gradient descent, penalties, augmented Lagrangians, Bundles, etc. In this paper, an optimization method for convex problems called ADMM has been discussed. The optimization of Lasso and Ridge regression using ADMM was part of our study, as well as a comparison of the ADMM implementation with the original model from Scikit-Learn's machine learning library and the gradient descent optimization approach for the same problems on the Diabetes dataset. We then compared these models using certain key evaluation metrics. Furthermore, we examined the performance of the three models and discovered that ADMM is an efficient algorithm as it converges to the optimum faster.**

***Index Terms*—Alternating Direction of Method of Multipliers(ADMM), Optimization, Lagrange multiplier, LASSO, Ridge, Convex, Gradient descent, SciKit-Learn machine learning library.**

I. Introduction

In general, ADMM is an algorithm for resolving some convex optimization problems [1]. It breaks the problems into two or more smaller chunks. ADMM finds the global convergence of any minimizing problem consisting of separable convex functions, given a constraint that is satisfied and a sufficiently small dual step [2]. This algorithm is like a modification of the augmented Lagrangian method that will be employing a partial update to all the dual variables. It is found in a lot of applications these days though it was present for the past few years. In the 1970s and 1980s, ADMM was extensively investigated as an alternative to penalty methods. The employment of ADMM in the area of data-distributed minimization on a huge scale has shown to be a good choice. ADMM is used mainly to speed up the convergence process. [3]

Our objective is to simplify the constrained optimization problem by breaking the primary loss function into smaller functions of two or more variables, then solving the functions individually, and finally coordinating local sub-problems to solve the global problem [4]. In technical terms, the core idea behind it is broadcasting and gathering, where broadcasting means splitting the (main) objective function into smaller parts and giving it to multiple processors for solving, and in gathering, the results of these processors are collected, combined, and used to determine the rule for the next iteration. The advantage of using this technique is that it has no constraints on the objective function other than convexity. It's not even necessary for the loss function to be differentiable.

This is used to solve problems involving optimization, such as :

$$minimize \;\; f(x) + g(x).$$

When ADMM is used, the before mentioned minimization problem may now be expressed as:

$$\min_{x} \;\; f(x) + g(z) \text{ subjected to } Ax + Bz = c$$

The constraint $Ax + Bz = c$ shows the relation between the variables x and z [5]. ADMM's basic objective is to loop through the following 3 steps :

1. The first step is to minimize the Lagrangian over x while keeping z and μ constant, with μ being the Lagrange multiplier.
2. The second step is to minimize the Lagrangian over z while keeping x and μ constant.
3. The third step is to use gradient ascent for updating the Lagrange multipliers μ.

Instead of doing multiple iterations until it converges, the algorithm will proceed directly to update the Lagrange multiplier (μ) and then repeat the process. Here we are using gradient ascent because at optima x and z becomes a ***saddle point***,

as we are trying to minimize the x and z and maximize the Lagrange multiplier (μ). So we can say that the ADMM will find a *saddle point*.

ADMM takes use of the method of multiplier's strong convergence features as well as the decomposability quality of dual ascent to solve optimization problems that are too big for generic optimisation solvers to handle [6].

Regularization methods are used to cope with overfitting and huge datasets. We know that there are two types of regularization techniques: Ridge and Lasso, which penalize the regression function [7]. We compare problems based on the gradient-descent technique and ADMM and show that the ADMM algorithm outperforms the gradient-ascent approach substantially [8].

II. Related Works

Glowinski and Marroco, Gabay and Mercier when collaborated in the mid-1970s, they first introduced ADMM, with roots dating back to the mid-1950s. Around the year 1974, the ADMM was created from an augmented method with a Lagrangian multiplier and has been known as ALG2 back then. In the year 1969, they were referred to as the Method of Multipliers [9]. A wide variety of applications of the ADMM have been explored in recent years, including Image Processing, Machine Learning, Statistics, Sparse Optimizations, Data Mining and other areas of relevance [10] [11]. ADMM has increased in popularity as an outcome of these implementations. Ever Since it had been called ALG2, it helped in solving numerous numerical problems related to Mechanics, Physics, Differential Geometry, etc.

In high dimensional problems, regularized optimization poses particular challenges since regularization is a natural mechanism for overcoming ill-posedness. As ADMM is efficient at solving regularized problems, it can act as an ideal candidate for stochastic optimization in higher dimensions. On the basis of ADMM, which has proven useful in distributed convex optimization, a comparable algorithm is devised [12]. Using Least Square denoising to enhance the ADMM based Hyperspectral Image Classification , was discussed in the paper [13].

This method dates back to the nineteen fifties, but it was developed after a span of 20 years [6], this ADMM algorithm is similar to other algorithms like Douglas-Rachford splitting, multiplier methods, Dykstra's alternating projections, dual decomposition, proximal methods like gradient descent, and so on [14]. The ADMM approach is used for L1 mean filtering and L1 variance filtering, which are important problems in signal processing with many other applications, including financial or biological data analysis,Setzer-2011 [15]. Before fitting a parametric model, data is usually pre-processed using mean and variance filters.

In 1986, the L1 penalty for both fitting and penalizing coefficients was independently developed and published in geophysics literature. R.Tibshirani introduced the LASSO in 1996, which adds an $l1$ penalty term to the least square loss in order to pick the relevant components and improve prediction accuracy [16]. Lasso analysis has been widely used, especially to analyze biological data, where there are only a small number of factors actually related to any outcome. And in the year 1970, in their publications "RIDGE regressions: a biased estimate of non-orthogonal issues" and "RIDGE regressions: applications in non-orthogonal problems," Hoerl and Kennard formulated the idea of Ridge regression.

III. Algorithm

As ADMM works on the idea of divide and conquers we first divide the constraint optimization problems into smaller functions and then find the solution to these individual parts. Because LASSO and Ridge problems include a likelihood component and $l1$, $l2$ penalty terms respectively, such a property of ADMM implies that it is suited for them, and by introducing a new variable, we may be able to fix the problems more easily [16] [17].

First, we discover x that minimizes the objective function with z and μ fixed, then we find z that minimizes the objective function with x and μ fixed, and lastly, we update μ.

ADMM was used to solve the LASSO penalized optimization problem. The regularization form of LASSO is as follows [18]:

$$\min_x \; l(x) + \lambda \left\|x\right\|_1$$

where $l(x)$ is the likelihood component, We now add a new variable z and modify the above LASSO problem as [19]

$$\min_x \; l(x) + \lambda \left\|z\right\|_1 \text{ subject to } x = z$$

And we apply the ADMM algorithm for LASSO regression as follows :

Algorithm 1 ADMM for solving LASSO problem

$x_0 \leftarrow 0$ ▷ initialize
$z_0 \leftarrow 0$
$\mu_0 \leftarrow 0$
$k \leftarrow 0$
while convergence criterion is not satisfied **do**
  $x^{k+1} \leftarrow (A^T A + \rho I)^{-1} \times (A^T b + \rho \times z^k - \mu^k)$
  $z^{k+1} \leftarrow S_{\lambda/\rho} \times (x^{k+1} + \mu^k)$
  $\mu^{k+1} \leftarrow \mu^k + \rho(x^{k+1} - z^{k+1})$
  if convergence criterion is satisfied **then**
    break
  else
    $k \leftarrow k + 1$
  end if
end while

Similarly To solve a ridge penalized problem we used ADMM, and Ridge regularization is of the form [18]:

$$\min_x \; l(x) + \lambda \left\|x\right\|_2^2$$

With an $l2$ penalized norm squared instead of an $l1$ norm,

where $l(x)$ is the likelihood component. We establish a new variable z as we did in LASSO, and now modify the above Ridge problem as

$$\min_x \ l(x) + \lambda \|z\|_2^2 \text{ subject to } x = z$$

And we apply the ADMM algorithm again for Ridge regression as follows:

Algorithm 2 ADMM for solving Ridge problem

```
x_0 ← 0                                              ▷ initialize
z_0 ← 0
μ_0 ← 0
k ← 0
while convergence criterion is not satisfied do
    x^{k+1} ← (A^T A + ρI)^{-1} × (A^T b + ρ × z^k − μ^k)
    z^{k+1} ← (ρ × x^{k+1} + μ^k)/(2 × λ + ρ)
    μ^{k+1} ← μ^k + ρ(x^{k+1} − z^{k+1})
    if convergence criterion is satisfied then
        break
    else
        k ← k + 1
    end if
end while
```

These are three main steps involved in the ADMM optimization, which are: the x update, z update and the multiplier μ update.

IV. Methodology

The ADMM has been widely researched for various linearly constrained convex problems with a primary objective of separating the target problem into distinct convex problems and then handling them separately with independent variables for each function or part, for example, we have a minimization problem is:

$$\min_x \ h(x)$$

Then to use ADMM , this above function is rewritten in the form of $g(x) + f(x)$, where $f(x)$ function and $g(x)$ function are separated from $h(x)$ so we can now write the problem as

$$\min_x \ f(x) + g(x)$$

And after removing the dependence or the use of the same variables in two functions we can change it as

$$\min_{x,z} \ f(x) + g(z) \text{ subject to } Ax + Bz = c$$

where $Ax + Bz = c$ is some relation between the two variables x and z. Despite the seeming triviality of this change, using the augmented Lagrangian method with a separable objective function in both x and z, the problem can now be tackled. This problem may be approximated using the ADMM approach by first solving for x with z fixed and μ fixed, then solving for z with x fixed and μ fixed. Instead of iterating until convergence, the method changes the dual variable immediately and continues repeating the procedure until a particular stopping criterion is satisfied.

The ADMM method then calculates the "saddle point" of the Augmented Lagrangian for the related problem .

The Below equation is the general Augmented Lagrangian for any optimization problem with a constraint.

$$L_\rho(x, z, \mu) = f(x) + g(z) + \mu^T(Ax + Bz - c) + \tfrac{\rho}{2} \|Ax + Bz - c\|_2^2$$

The algorithm now minimizes the $L_\rho(x, z, \mu)$ function with respect to x for fixed z and μ, then minimizes z for fixed x and μ, and finally performs a gradient step with respect to μ for fixed x and z.

$$x^{k+1} = \arg\min \ L_\rho(x, z^k, \mu^k)$$
$$z^{k+1} = \arg\min \ L_\rho(x^k, z, \mu^k)$$
$$\mu^{k+1} = \mu^k + \rho(Ax^{k+1} + Bz^{k+1} - c)$$

However, this only works if $f(x)+g(z)$ is a convex function, which is dependent on two assumptions: There exists an optimal solution that will minimize the $f(x) + g(z)$ function and that solution **(x*,z*)** satisfies the constraint such that $(Ax^* + Bz^* - c = 0)$ [20] [9].

ADMM takes use of the method of multipliers' strong convergence features as well as the decomposability quality of dual ascent to solve optimization problems that are too big for generic optimization solvers to handle [6].

A. Lasso using ADMM:

ADMM is a suitable option for lasso problems since it divides target problems into two different functions and solves each individually. The structure of the $l1$ penalty in the lasso problems provideing improved performance in variable selection since it constrains the solution to be sparse [16]. The lasso problem involves solving

$$\arg\min_x \ l(x) + \lambda \|x\|_1$$

Since $l(x)$ is a convex function and $l(x) = \frac{1}{2}\|Ax - b\|_2^2$, the optimization problem may now be reformulated as:

$$\arg\min_x \ \tfrac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_1$$

And λ being some regularization constant that is usually chosen from cross-validation. In ADMM form, this may be split into 2 or more functions, $g(x)$ and $f(x)$, and the ADMM structure is:

$$\arg\min_{x,z} \ f(x) + g(z) \text{ subject to } x - z = 0$$

where $f(x) = \frac{1}{2}\|AX - b\|_2^2$ and $g(z) = \lambda \|z\|_1$.

$$\arg\min_{x,z} \ \tfrac{1}{2}\|AX - b\|_2^2 + \lambda \|z\|_1 \text{ subject to } x - z = 0$$

While applying the ADMM approach to the lasso problem, the 3 major phases are: Firstly, minimization of x, second, minimization of z and then the updation of the dual variable

μ [21]. The minimization of x is done as follows:

Firstly for the minimization of the x we do

$$x^{k+1} = \arg\min_x \ \tfrac{1}{2}\|Ax-b\|_2^2 + \tfrac{\rho}{2}\left\|x - z^k + \tfrac{1}{\rho}\mu^k\right\|_2^2$$

And, following the final simplification of the above equation, it will appear as:

$$x^{k+1} = (A^TA + \rho I)^{-1}(A^Tb + \rho(z^k) - \mu^k)$$

Now for the minimization of z,

$$z^{k+1} = \arg\min_z \ \lambda\|z\|_1 + \tfrac{\rho}{2}\left\|x^{k+1} - z + \tfrac{1}{\rho}\mu^k\right\|_2^2$$

So z update can be written as $z^{k+1} = S_{\lambda/\rho}(x^{k+1} + \frac{\mu^k}{\rho})$, [19] where $S_{\lambda/\rho}$ represents a soft threshold operator. Finally, we simplify the above equation as

$$z^{k+1} = x^{k+1} + \frac{\mu^k}{\rho} - \frac{\lambda}{\rho}sign(z)$$

Here the $sign(z)$ represents a signum function. And the dual variable μ is updated in the last step :

$$\mu^{k+1} = \mu^k + \rho(x^{k+1} - z^{k+1})$$

This is like a dual ascent along the variable μ with the learning rate being ρ. And from the method of multipliers, we know that ρ, the penalty factor, is chosen as the learning rate for this gradient ascent with $\rho > 0$. For the dual update, we will find the gradient of the penalty function which is of form

$$P = \mu^k(x^{k+1} - z^{k+1}) + \tfrac{\rho}{2}(x^{k+1} - z^{k+1})^2$$

Since ρ is always greater than zero $(A^TA + \rho I)$ is always a non-singular and square matrix. If the penalty factor(ρ) in the augmented Lagrangian method is selected to be sufficiently big, ADMM will be converging to a set of optimal solutions [22]. The x update is just ridge regression. So, ADMM on lasso can be viewed as an iterative form of ridge regression.

B. Ridge using ADMM:

The Ridge Regression objective is similar to the LASSO objective, but instead of using the $l1$ norm as a regularization term, we use the $l2$ norm squared in our analysis along with the λ. This involves solving [23]

$$\arg\min_x \quad l(x) + \lambda\|x\|_2^2$$

So the optimization problem using the ADMM looks like this:

$$\arg\min_{x,z} \quad l(x) + \lambda\|z\|_2^2 \text{ subject to } x - z = 0$$

with $l(x)$ being the loss function and $l(x) = \frac{1}{2}\|Ax-b\|_2^2$. In an extended form, the basic ridge regression optimization is as follows:

$$\arg\min_{x,z} \quad \tfrac{1}{2}\|Ax-b\|_2^2 + \lambda\|z\|_2^2$$

Where x is the solution vector for the series of input features represented in a matrix form A and b is the vector of all the respective outputs. Similar to lasso this ridge regression can be implemented using ADMM with 3 main steps: Firstly, minimization of x, second, minimization of z and then the updation of the dual variable μ. The minimization of x is done as follows:

$$x^{k+1} = \arg\min_x \ \tfrac{1}{2}\|Ax-b\|_2^2 + \tfrac{\rho}{2}\left\|x - z^k + \tfrac{1}{\rho}\mu^k\right\|_2^2$$

And after the final simplification of the equation, it looks like

$$x^{k+1} = (A^TA + \rho I)^{-1}(A^Tb + \rho(z^k - \mu^k))$$

Also, the minimization of z is shown as:

$$z^{k+1} = \arg\min_z \quad \lambda\|z\|_2^2 \ + \tfrac{\rho}{2}\left\|x^{k+1} - z + \tfrac{1}{\rho}\mu^k\right\|_2^2$$

which can now be simplified to

$$z^{k+1} = \frac{\rho x^{k+1} + \mu^k}{2\lambda + \rho}$$

The last step in the ADMM method is to update the dual variable, it is updated as:

$$\mu^{k+1} = \mu^k + \rho(x^{k+1} - z^{k+1})$$

These three steps are repeated until a convergence threshold is attained. $f(x^k) \to f^*$ and the primal condition $x^k - z^k \to 0$ as $k \to \infty$ and f^* being the optimum. In reality, if ρ is chosen well, we can increase the convergence rate and obtain the solution with fewer iterations. We need to iterate many times if it is not well chosen. [15]

And the criterion is that when the norm of primal residual and norm of dual residual values become smaller than feasibility tolerances for the primal and dual feasibility conditions respectively. The primal residual iterate and dual residual iterate at a patricular iteration k are formulated as

$$e_p^k = (x^k - z^k), \quad e_d^k = -\rho(z^k - z^{k-1})$$

The feasibility tolerances say $e^{pri} > 0 e^{dual} > 0$ can be set using some abstol and reltol .

$$\epsilon^{\text{pri}} = \sqrt{n}\epsilon^{\text{abs}} + \epsilon^{\text{rel}}\max\{\|x^k\|_2, \|z^k\|_2\},$$
$$\epsilon^{\text{dual}} = \sqrt{n}\epsilon^{\text{abs}} + \epsilon^{\text{rel}}\rho\|u^k\|_2,$$

where $e^{abs} > 0$ and $e^{rel} > 0$ are tolerances with e^{abs} the absolute tolerance and e^{rel} being the relative tolerance, [14] [15].

Now after updating in both the regression techniques - LASSO and Ridge, we now obtain the final optimal solutions.

V. DATASET

The study of LASSO and Ridge using ADMM(alternating method of multipliers) is done based on their cost function values plotted against the number of iterations. In addition, the dataset used is a diabetes dataset acquired from Medical City Hospital in Baghdad. The dataset comprised of medical information for which analysis was done on patients in laboratories. Our initial data was identified to be Patient number, Gender, Blood Sugar level, Age, Body mass index (BMI), Urea, Cholesterol (Chol), Creatinine ratio (Cr), and HbA1c, including total cholesterol, LDL cholesterol, and

VLDL cholesterol, triglycerides(TG) and HDL cholesterol, and Diabetic, Predictive Diabetes, or Non-Diabetic Diabetes Class (the patient's diabetes status) [24]. This dataset consists of details of 1000 patients and covers three classes- Diabetic(Y), Predictive Diabetes(P), or Non-Diabetic Diabetes(N). So this dataset is of size (1000x14).

Some Manipulations on the Dataset are done to remove the categorical features like the "Gender" feature and make them into numerical values, and we dropped the features like "No_Patient" and "ID" as they do not contribute to the "Sugar_Level" feature of a particular patient. So the final dataset now has 12 features - **AGE, Urea, Cr, Chol, TG, HDL, LDL, VLDL, BMI, Sugar_Level, Gender_M, CLASS**. The "Gender_M" is used to represent the gender of a patient assuming a binary value (0 or 1), If the patient is male then the" Gender_M" value is 1 else 0.

VI. Experimentation

	AGE	Urea	Cr	Chol	TG	HDL	LDL	VLDL	BMI	Sugar_Level	Gender_M
AGE	1.000000	0.105092	0.054941	0.036649	0.148204	-0.020038	0.016105	-0.087903	0.375956	0.379136	0.021486
Urea	0.105092	1.000000	0.624134	0.001852	0.040980	-0.036994	-0.007301	-0.011191	0.045618	-0.023603	0.116311
Cr	0.054941	0.624134	1.000000	-0.007097	0.056579	-0.023804	0.039479	0.009615	0.054746	-0.037412	0.154870
Chol	0.036649	0.001852	-0.007097	1.000000	0.321789	0.103814	0.416665	0.076294	0.013678	0.177489	-0.064763
TG	0.148204	0.040980	0.056579	0.321789	1.000000	-0.083001	0.015378	0.144570	0.110757	0.218556	0.052111
HDL	-0.020038	-0.036994	-0.023804	0.103814	-0.083001	1.000000	-0.142079	-0.059275	0.072409	0.028933	-0.130130
LDL	0.016105	-0.007301	0.039479	0.416665	0.015378	-0.142079	1.000000	0.062795	-0.067322	0.011057	0.054563
VLDL	-0.087903	-0.011191	0.009615	0.076294	0.144570	-0.059275	0.062795	1.000000	0.198133	0.073462	0.194120
BMI	0.375956	0.045618	0.054746	0.013678	0.110757	0.072409	-0.067322	0.198133	1.000000	0.413350	0.072097
Sugar_Level	0.379136	-0.023603	-0.037412	0.177489	0.218556	0.028933	0.011057	0.073462	0.413350	1.000000	-0.009362
Gender_M	0.021486	0.116311	0.154870	-0.064763	0.052111	-0.130130	0.054563	0.194120	0.072097	-0.009362	1.000000

Fig. 1. Co-Relation among the features of the final dataset

The features that contribute the most to the "Sugar_Level" are extracted by computing the correlation among the features. And the one which has the highest correlation among all of the features is selected for optimization. The feature "BMI" is much more correlated compared to the others. Now the lasso model for this dataset is optimized using ADMM, finding the optimal solution vector consisting of the coefficients. And as the data is split into train and test parts, we predicted the "sugar level" values for the test data using the newly obtained optimal solution. And finally, the solution obtained from this is compared with that of SciKit-Learn's inbuilt Lasso model along with the solution obtained from a gradient descent optimization algorithm for the modified Diabetes dataset. A similar process is done for Ridge regression to find the optimal solution from ridge using ADMM and the final result computed using the optimal solution is now compared with SciKit-Learn's inbuilt Ridge model along with the solution obtained from a gradient descent optimization algorithm for the same dataset.

Also, from the iteration history which consists of r_norm, s_norm, eps_pri, eps_dual, objval [25] some iterates such as (r_norm, s_norm, objval, time) vs iteration are plotted for both the problems.

objval: Object (cost) function value.
r_norm: Norm of primal residual.
s_norm: Norm of dual residual.
eps_pri: Feasibility tolerance for primal feasibility condition.
eps_dual: Feasibility tolerance for dual feasibility condition.

The time taken for each iteration in Gradient Descent approach has been represented graphically. Additionally, several evaluation criteria such as the Mean Absolute Error, Standard Error, total time taken etc. are computed and compared.

VII. Results & Discussions

The results for the two problems LASSO and Ridge from the three models are compared using Mean Absolute Error,the Standard error and (RMSE)Root Mean Square Error of the predicted BMI and the test BMI values. The time taken and the objective function value plots vs each iteration plots are shown.

Also, the plot of the test data vs the predicted values from the LASSO SciKit-Learn model, Gradient Descent model and LASSO using the ADMM model is done. In the same way graphs for the Ridge regression are obtained. And the results depict that the graphs for the Gradient Descent vs ADMM are almost the same with negligible error. From the SciKit-Learn vs ADMM graphs we may see a slight difference in the graph with minimum error.

A. Lasso Regression

We ran our Solvers for the LASSO using ADMM, sklearn's LASSO model and the LASSO optimization using Gradient Descent on the Diabetes Dataset, the final objective history is plot vs the iteration number.

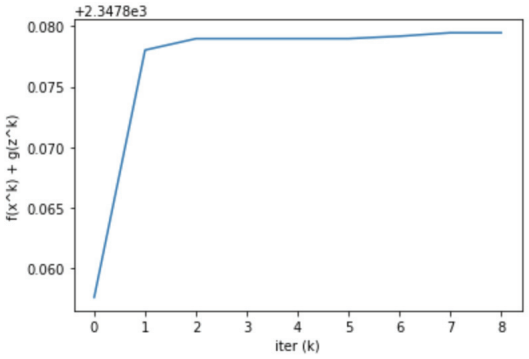

Fig. 2. LASSO Objective function for gradient ascent step for multipliers

We have kept track of the time that takes for each iteration, so when we plot the time vs iteration graph, the time will reduce with each iteration as it converges to the optimal solution. We recorded the iteration history and it consists of the "r_norm" and "s_norm" which are the norms of the primal and dual residual respectively.

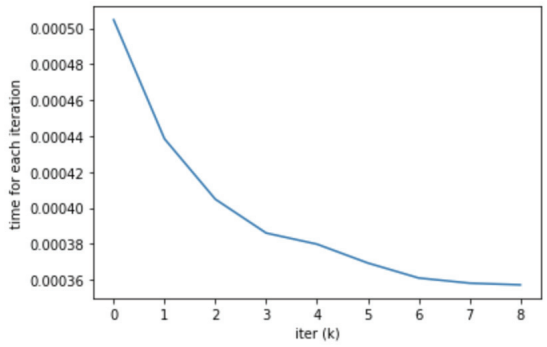

Fig. 3. Time vs Iteration for the LASSO-ADMM implementation

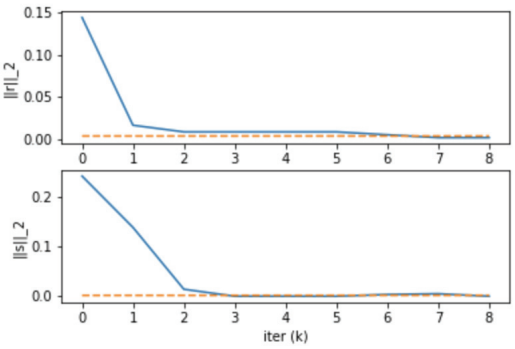

Fig. 4. The respective r_norm and the s_norm values vs the iteration for LASSO-ADMM

As shown in Fig3. The graph of time versus iteration shows that the graph decreases as we see a rise in number of iterations, and as it converges, the time spent on each iteration decreases. And in the Fig4. When convergence occurs, r_norm and s_norm values also decrease, which means the difference between the updates will decrease.

And below is a graph of the time vs the iteration for the Gradient Descent implementation of LASSO regression,

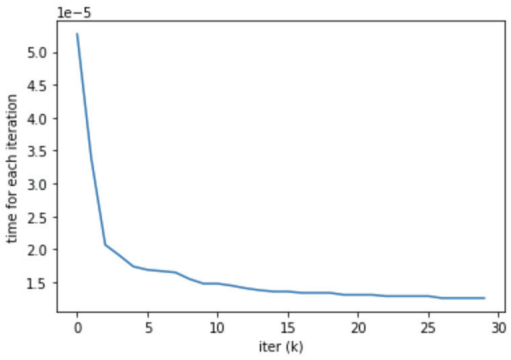

Fig. 5. Time vs Iteration for the LASSO-GD implementation

We see that , Gradient Descent(GD) algorithm converges to the optimal solution after 30 iterations for the LASSO problem.

B. Ridge Regression

We ran our Solvers for the Ridge using ADMM, sklearn's Ridge model and the Ridge optimization using Gradient Descent on the Diabetes Dataset, the final objective history is plot vs the numbers of iterations.

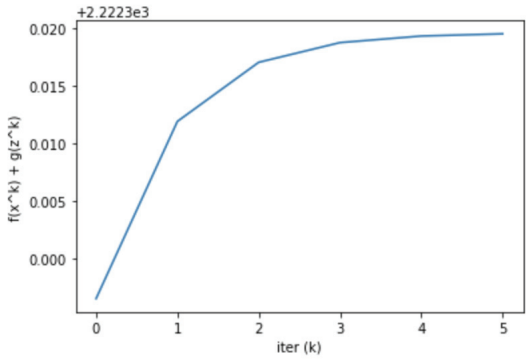

Fig. 6. Ridge Objective function for gradient ascent step for multipliers

Similar to LASSO using ADMM we kept track of the iteration history for the Ridge using ADMM so we again plot the time vs iteration graph, A similar pattern could be seen where the graph has a negative slope and decreases as the number of iterations increased.

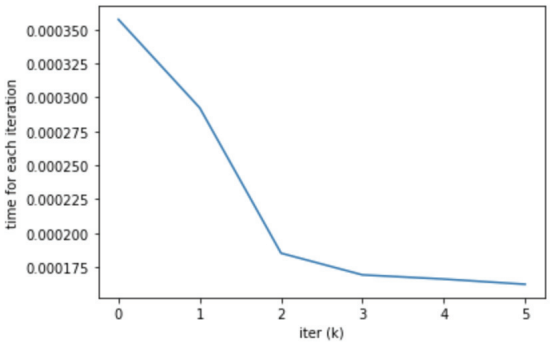

Fig. 7. Time vs Iteration for the Ridge-ADMM implementation

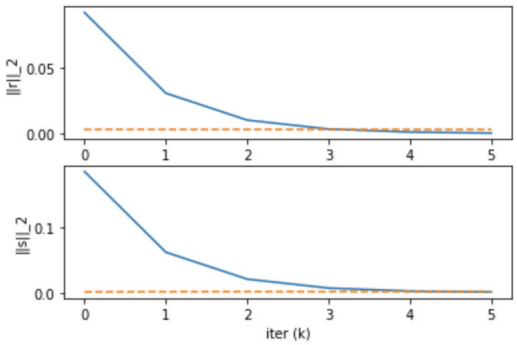

Fig. 8. The respective r_norm and the s_norm values vs the iteration for Ridge-ADMM

Fig 8. is a graph showing the plots of ”r_norm” values and the ”s_norm” vs the number of iterations.

And below is a graph of the time vs the iteration for the Gradient Descent implementation of Ridge regression,

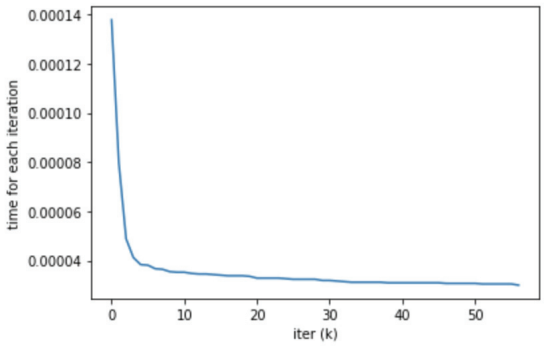

Fig. 9. Time vs Iteration for the LASSO-GD implementation

We see that Gradient Descent(GD) algorithm converges to the optimal solution after 57 iterations for the Ridge problem.

C. Comparisons

ADMM, Scikit-Learn models and optimization using Gradient Descent were analysed using certain key evaluation metrics such as RMSE - the square root of the average of the squares of difference of all the real and estimated data points, MAE - it is the average of the differences of the real and estimated data points, and Standard Error - a method used to estimate how the values in the dataset are spread out. In addition, the time required for all three models to converge to the optimal solution is determined and compared.

TABLE I
PERFORMANCE EVALUATION OF ADMM MODEL WITH OTHER MODELS

Model	RMSE	MAE	Std.Error	Time
LASSO using ADMM	2.2204	1.7872	0.0952	0.00110 s
LASSO using Scikit - Learn	2.2564	1.8425	0.0639	0.00577 s
LASSO using GD	2.2262	1.8016	0.0855	0.00140 s
Ridge using ADMM	2.2249	1.7789	0.0947	0.00177 s
Ridge using Scikit - Learn	2.2480	1.8340	0.0673	0.00486 s
Ridge using GD	2.2287	1.7983	0.0878	0.00273 s

From the above tabular representation, we see that the time required for obtaining an optimal solution is lower for ADMM when compared to Scikit-Learn’s model and optimization using Gradient Descent, because we are splitting the main loss function in ADMM, and as a result, the time required for the ADMM implementation is shorter. The error for the ADMM solver is lesser than the error for the Scikit-Learn solver and optimization using Gradient Descent when comparing predicted and test data. Still, there is not much difference between them. Finally, we can conclude that the use of ADMM to solve optimization problems is indeed an effective approach.

As we have compared the results for the LASSO-ADMM model and the Scikit-Learn LASSO model, the below graphical representation is a comparison for the LASSO implementation using the ADMM Solver and the Sklearn LASSO Solver.

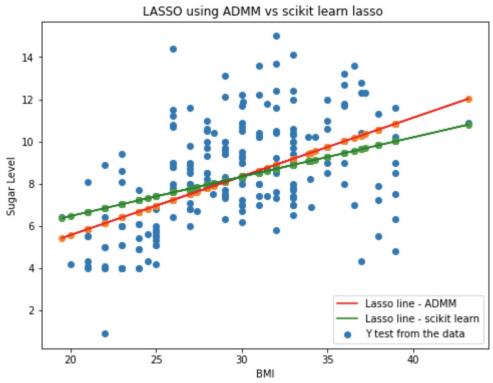

Fig. 10. Comparison with Scikit-Learn’s Prediction

And Fig.11. is a comparison for the LASSO implementation using the ADMM solver and the optimization using Gradient Descent. We see that the predicted lines from both the algorithms almost coincide with each other.

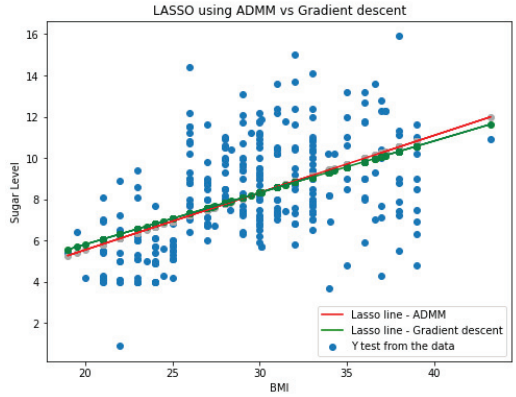

Fig. 11. Comparison with optimization using Gradient Descent

Below is a comparison for the Ridge implementation using the ADMM Solver and the Scikit-Learn Ridge Solver.

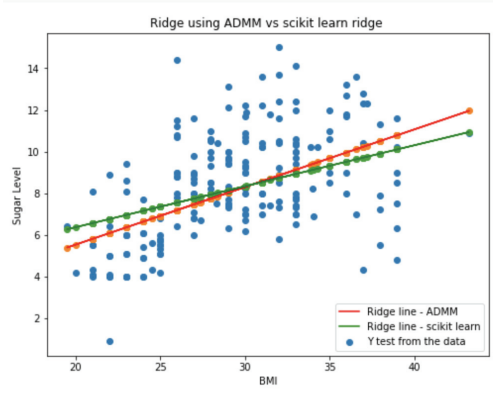

Fig. 12. Comparison with Scikit-Learn’s Prediction

Fig.13 is a comparison for the Ridge implementation using ADMM Solver and optimization using Gradient Descent.

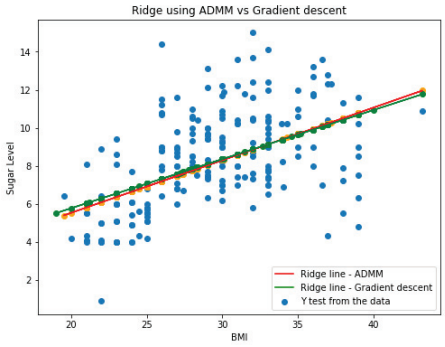

Fig. 13. Comparison with optimization using Gradient Descent

VIII. Conclusion

In this paper, we implemented two regression algorithms namely LASSO and Ridge, based on ADMM and optimised their cost functions to obtain optimal solutions. For each of the implementations (ADMM, Gradient Descent and Scikit-Learn), we calculated the time taken. As a result, ADMM performed faster than gradient descent and SciKit-Learn and also the solution obtained from ADMM implementation was better than that of Scikit-Learn's solver and gradient descent method. Furthermore, we observed that the gradient descent algorithm obtained the optimal solution after 30 iterations for the LASSO problem, whereas the ADMM implementation took 8 iterations. When using the gradient descent method to solve the Ridge problem, the optimal solution was obtained after 57 iterations, whereas using ADMM, it took 5 iterations. It shows how efficiently the ADMM implementation solves optimization problems. ADMM implementation on multi-threaded environment in a single machine takes longer to converge, ADMM helps in getting faster convergence in distributed settings especially when we are adressing a convex optimization problem on large dataset. The evaluation metrics are computed for the three implementations. We see that the difference between the gradient descent and the ADMM implementations is negligible or trivial. Comparing the SciKit-Learn implementation with the ADMM implementation we see a slight difference between the evaluation metrics. Therefore, we can conclude that ADMM converges quickly to the optimum point.

References

[1] Harikumar, "Admm based algorithm for spike and smooth signal separation using over-complete dictionary. 2015 international conference on communications and signal processing (iccsp). ieee, 2015."

[2] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *Mathematical Programming*, vol. 162, no. 1, pp. 165–199, 2017.

[3] S. e. a. Srivatsa, "Application of least square denoising to improve admm based hyperspectral image classification. procedia computer science 93 (2016): 416-423."

[4] S. P. Boyd, "Admm," Dec. 2021. [Online]. Available: https://stanford.edu/~boyd/admm.html

[5] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. Jordan, "A general analysis of the convergence of admm," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 343–352. [Online]. Available: https://proceedings.mlr.press/v37/nishihara15.html

[6] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (admm): Quadratic problems," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 644–658, 2015.

[7] P. Tejaswini and P. Sivraj., "Artificial intelligence based state of charge estimation of li-ion battery for ev applications. 2020 5th international conference on communication and electronics systems (icces). ieee, 2020."

[8] P. Šulc, S. Backhaus, and M. Chertkov, "Optimal distributed control of reactive power via the alternating direction method of multipliers," *IEEE Transactions on Energy Conversion*, vol. 29, no. 4, pp. 968–977, 2014.

[9] L. Li, X. Wang, and G. Wang, "Alternating direction method of multipliers for separable convex optimization of real functions in complex variables," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[10] Wikipedia contributors, "Augmented lagrangian method — Wikipedia, the free encyclopedia," 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Augmented_Lagrangian_method&oldid=1052289126

[11] V. Ranganathan and K. A. Narayanankutty., "Admm based lms algorithm for constant modulus signals in adaptive beamforming. 2017 14th ieee india council international conference (indicon). ieee, 2017."

[12] M. Luo, L. Zhang, J. Liu, J. Guo, and Q. Zheng, "Distributed extreme learning machine with alternating direction method of multiplier," *Neurocomputing*, vol. 261, pp. 164–170, 2017.

[13] e. a. Srivatsa, S., "Application of least square denoising to improve admm based hyperspectral image classification. procedia computer science, vol. 93, jan. 2016, pp. 416–23. sciencedirect, https://doi.org/10.1016/j.procs.2016.07.228."

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[15] B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang, "An admm algorithm for a class of total variation regularized estimation problems*," *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 83–88, 2012, 16th IFAC Symposium on System Identification. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667015379325

[16] Y. He, "The application of alternating direction method of multipliers on l1-norms problems," in *Journal of Physics: Conference Series*, vol. 1187, no. 4. IOP Publishing, 2019, p. 042070.

[17] V. S. Swarna, M. and K. P. Soman., "Effect of dimensionality reduction on sparsity based hyperspectral unmixing. international conference on soft computing and pattern recognition. springer, cham, 2016."

[18] L. Melkumova and S. Shatskikh, "Comparing ridge and lasso estimators for data analysis," *Procedia Engineering*, vol. 201, pp. 746–755, 2017, 3rd International Conference "Information Technology and Nanotechnology", ITNT-2017, 25-27 April 2017, Samara, Russia. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877705817341474

[19] F. Ma, M. Ni, X. Zhang, and Z. Yu, "Solving lasso: Extended admm is more efficient than admm," in *2015 Chinese Automation Congress (CAC)*, 2015, pp. 55–58.

[20] A. Aich., "A brief review of alternating direction method of multipliers (admm). admm_review.pdf, https://abhishekaich27.github.io/data/writeups/admm_review.pdf."

[21] "Math 301: Advanced topics in convex optimization. no. lecture 26 — march 9, winter 2015. consensus.pdf, https://candes.su.domains/teaching/math301/lectures/consensus.pdf. author=Hamid Javadi, and Emmanuel Candes."

[22] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 337–364, 2016.

[23] M. Humayoo and X. Cheng, "Parameter estimation with the ordered 2 regularization via an alternating direction method of multipliers," *Applied Sciences*, vol. 9, no. 20, p. 4291, 2019.

[24] A. Rashid, "Diabetes dataset," *Mendeley Data*, vol. 1, 2020.

[25] K. You and X. Zhu, "Algorithms using alternating direction method of multipliers [r package admm version 0.3.3," *Comprehensive R Archive Network (CRAN)*, 2021.