Fractional Gradient Based Optimization for Nonlinear Separable Data

D P Hapsari[1], M F Rozi[2]
[1]Department of Informatic Engineering, Institut Teknologi Adhi Tama Surabaya (ITATS), Surabaya
[2]Department of Information System, Institut Adhi Tama Surabaya (ITATS), Surabaya

Email: [1]m.fahrur.rozi@itats.ac.id, [2]dian.puspita@itats.ac.id

Received: 2022-03-07 Received in revised from 2022-03-22 Accepted: 2022-03-28

Abstrak

Support Vector Machine atau SVM classifier merupakan salah satu algoritma machine learning yang bertugas memprediksi data. Pengklasifikasi tradisional memiliki keterbatasan dalam proses pelatihan data skala besar, cenderung lambat. Penelitian ini bertujuan untuk meningkatkan efisiensi pengklasifikasi SVM menggunakan algoritma optimasi penurunan gradien fraksional, sehingga kecepatan proses pelatihan data dapat ditingkatkan saat menggunakan data skala besar. Terdapat sepuluh kumpulan data numerik yang digunakan dalam simulasi yang digunakan untuk menguji kinerja classifier SVM yang telah dioptimasi menggunakan algoritma fractional gradient descent tipe Caputo. Dalam makalah ini, kami menggunakan rumus turunan Caputo untuk menghitung penurunan gradien orde pecahan dari fungsi kesalahan terhadap bobot dan memperoleh konvergensi deterministik untuk meningkatkan kecepatan konvergensi turunan orde pecahan tipe Caputo. Hasil pengujian menunjukkan bahwa pengklasifikasi SVM yang dioptimalkan mencapai waktu konvergensi yang lebih cepat dengan iterasi dan nilai kesalahan yang kecil. Untuk penelitian selanjutnya, pengklasifikasi linier SVM yang dioptimalkan dengan penurunan gradien fraksional diimplementasikan pada masalah data kelas yang tidak seimbang.

Kata Kunci: *SVM Classifier, Fractional Gradient Based, Nonlinear Separable Data*

Abstract

The Support Vector Machine or SVM classifier is one of the machine learning algorithms whose job is to predict data. Traditional classifier has limitations in the process of training large-scale data, tends to be slow. This study aims to increase the efficiency of the SVM classifier using a fractional gradient descent optimization algorithm, so that the speed of the data training process can be increased when using large-scale data. There are ten numerical data sets used in the simulation that are used to test the performance of the SVM classifier that has been optimized using the Caputo type fractional gradient descent algorithm. In this paper, we use the Caputo derivative formula to calculate the fractional-order gradient descent from the error function with respect to weights and obtain a deterministic convergence to increase the speed of the Caputo type fractional-order derivative convergence. The test results show that the optimized SVM classifier achieves a faster convergence time with iterations and a small error value. For further research, the optimized SVM linear classifier with fractional gradient descent is implemented on the problem of unbalanced class data.

Keywords: SVM Classifier, Fractional Gradient Based, Nonlinear Separable Data

1. Introduction (bold, style = Heading 1)

In machine learning, using large-scale data to recognize patterns in the data that can be used for predictive activities. Supervised learning or supervised learning is one of the methods in machine learning that divides data into training data and test data. By doing learning on large-scale training data, it is hoped that you will get a learning model that will be used on the test data. This learning model will be used on new data and it is hoped that the model can make predictions [1]. The supervised learning method consists of several algorithms, including; Nearest Neighbour, Naive Bayes, Decision Trees,

Linear Regression, Logistic Regression, Support Vector Machines and Neural Networks. Not all of these algorithms are capable of processing large-scale data, some of which have the ability to process large-scale data takes a long time [2].

In large-scale data training activities, SVM as a linear classification method faces challenges with sluggish convergence and a little number of data [3][4]. SVM also has a flaw in that identifying the ideal parameters is challenging [5]. As a result, convex optimization is used to address the SVM linear classifier's optimization problem, which is directly related to the large-scale data training process. Time-consuming matrix operations are occasionally required in large-scale training approaches. This has to do with the size of the matrix, which keeps growing or becomes unworkable due to memory restrictions [6]. The large-scale data training of the SVM linear classifier is a convex optimization problem that scales with the size of the training set rather than the dimensions of the feature space. Data training efforts on a broad scale will become impractical as a result of this [7].

An unconstrained optimization technique is required for SVM classifiers for information preparation forms. The gradient-based unconstrained optimization technique is especially successful in terms of computing time speed for large-scale information sets, according to the research [8][9]. There have been a number studies on how to improve the gradient-based SVM demonstration, including optimization using the stochastic gradient descent approach. Furthermore, sub-gradient descent is included in optimization algorithms for gradient-based SVM models, which makes descent easier [10]. When dealing with huge data sets, both methodologies have been found to give substantial advantages over traditional approaches. There is a fractional gradient descent optimization method for optimization without constraints other than stochastic gradient descent. Fractional gradient descent is a constraint-free optimization method that has been shown to solve large-scale training optimization problems for linear classifier algorithms like neural networks [11].

Many studies using fractional-order derivatives have been carried out, several types of fractional-order derivatives other than Caputo include the Riemann–Liouville type fractional-order, the Grünwald–Letnikov type fractional-order, and the Atangana-Baleanu type fractional-order which is a fractional-order derivative type. which is widely used. For the problem of convergence and convergence speed, it is solved by using a fractional-order derivative of the Caputo type [12][13]. In this paper, using the Caputo derivative formula to calculate the fractional-order gradient descent of the error function with respect to weights and obtaining deterministic convergence to increase the convergence speed of the Caputo-type fractional-order derivative has more applications in physical processes and engineering problems [14][15].

This paper has the following research questions: First: how does the proposed fractional gradient-based optimization method affect the improvement of the convergence time, the improvement of the error value and the number of iterations in the training data activity? Second: how to determine the learning rate and fractional order parameters given for optimizing the convergence time, improving the error value and the number of iterations in training data activities? There are ten numerical data sets used in the simulation that are used to test the performance of the SVM classifier that has been optimized using the Caputo type fractional gradient descent algorithm.

1.1. Support Vector Machine Classifier Problem

The SVM classifier solves optimization problems using an algorithm or solver with the purpose of estimating the values of w and b. We can use a quadratic solver for the first stage of reducing the quadratic function that is subject to linear constraints. The problem is that the solver is inefficient when dealing with big amounts of data, so we use an alternative approach. The convex function can be minimized.

$$f(w,b) = \frac{1}{2}\sum_{j=1}^{d}\left(w^{(j)}\right)^2 + C\sum_{i=1}^{n}\max\{0,1 - y_i\,(w\cdot x_i + b)\} \quad (1)$$

The convex function's gradient descent is fairly simple; half is the sum of all the coordinates over all the dimensions of the square of the value of w. Calculating (j) takes $O(n)$ time, which is an

issue. Optimization based gradient especially stochastic gradient descent is an important approach to very large-scale machine learning problems and challenges for which exact gradients are difficult to calculate. This shows that stochastic gradient descent has a sub-linear convergence level of $O(\frac{1}{t})$ in the case of highly convex and $\mathcal{O}\left(\frac{1}{\sqrt{t}}\right)$ general convex classifier objective functions, where t is the number of iterations. This will be the basis of the algorithm that will be offered in this paper. General optimization problem formulation, start from the optimization problem in the following standard form:

Minimizing $f_0(x)$ with constraint $f_i(x) \leq 0, i = 1, \ldots, m$

$h_i(x) = 0, i = 1, \ldots, p$

with $f_i, h_i: R^n \rightarrow R$; x is the optimization variable;

f_0 is an objective function or a cost function; $f_i(x) \leq 0$ is the inequality constraint. Geometrically, this problem is concerned with minimizing f_0, over a set described as the intersection of the sublevel-0 set of f_i, {where the region is described by the solution set-0 of $h_i, \forall i$.

The feasible set C is the set of all feasible points, and the problem is feasible if there is a viable point. The problem is said to be infinite if $m = p = 0$. The optimal value is denoted by $f^* = \inf_{x \in C} f_0(x)$, and $f^* = +\infty$ if the problem is not feasible. A point $x \in C$ is an optimal point if $f(x) = f^*$ and the optimal set is $X_{opt} = \{x \in C | f(x) = f^*\}$ implicitly the constraint can be expressed as: $x \in \text{dom } f_i, x \in \text{dom } h_i$, which must be in a set $D = \text{dom } f_0 \cap \text{dom } f_m \cap \text{dom } h_1 \cap \ldots \cap \text{dom } h_p$ called the problem domain.

A feasible problem is a special case of the standard problem, which is a search for any feasible point. Then the real problem is look for $x \in C$ or specify $C = \emptyset$.

An optimization problem in standard form is a convex optimization problem if $f_0, f_1, \ldots, f_m$ are all convex, and h_i are all affine:

Minimizing $f_0(x)$ with constraints $f_i(x) \leq 0, i = 1, \ldots, m$

$a_i^T x - b_i = 0, i = 1, \ldots, p.$ (2)

This problem is often written in the form

Minimizing $f_0(x)$ with constraints $f_i(x) \leq 0, i = 1, \ldots, m$

$Ax = b$ with $A \in R^{p \times n}$ and $b = R^p$ (3)

The convex optimization problem has three key characteristics that distinguish it from non-convex optimization problems:

There are no local minima because every local optimum is also a global optimum.

Uncertain impropriety detection: the algorithm is easier to determine using the duality theorem.

When deciding really large issues, the numerical solution method is effective.

To understand the global optimality in convex problems, consider that $x \in C$ is a local optimal if it satisfies

$y \in C, \|y - x\| \leq R \rightarrow f_0(y) \geq f_0(x)$ (4)

for a $R > 0$. A point $x \in C$ global optimal means that $y \in C \rightarrow f_0(y) \geq f_0(x)$

For the convex optimization problem, any local solution is also a global solution. This can be proven: Suppose x the local optimal, but exists $y \in C$, with $f_0(y) \geq f_0(x)$. Then we can take a small step from x to y is $z = \lambda y + (1 - \lambda)x$ with a small $\lambda = 0$. Then z is close to , with $f_0(z) \geq f_0(x)$ which contradicts the local optimal.

There is also a first order condition to determine the optimization of the convex optimization problem. Suppose f_0 it is differentiable, then $x \in C$ is optimal if and only if

$y \in C \rightarrow \nabla f_0(x)^T (y - x) \geq 0.$ (5)

Thus $-\nabla f_0(x)$ defines the hyper support plane for C at x. That is, if it moves from along the x worth to y another, f_0 it doesn't go down.

2. Method

Caputo's Derivative Fractional. There are several definitions used for fractional derivatives. The three most common definitions of fractional calculus are Grunwald-Letnikov (GL), Riemann-Liouville

(RL), and Caputo. Caputo's fractional-order derivative. The definition of Caputo's fractional-order derivative of order is defined as follows.

$$ {}_{a}^{Caputo}D_t^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \int_a^t (t-\tau)^{n-\alpha-1} f^{(n)}(\tau) d\tau , \quad (6) $$

where ${}_{a}^{Caputo}D_t^{\alpha}$ *is a Caputo derivative operator,* α is fraksional order.

Because the initial values of fractional differential equations with Caputo derivatives and integer differential equations are the same, these derivatives can be used to solve a wide range of physical and engineering issues. In this study, only used Caputo's fractional order derivative to evaluate the SVM training algorithm for gradient-based optimization with fractional-order. Used the idea ${}_{a}D_t^{\alpha}$ of showing Caputo's fractional-order operator. Fractional order calculus is a natural generalization of classical integer calculus. The flow of the SVM Fractional Gradient Descent algorithm begins with;

Given a training set $S = \{(x_i, y_i)\}, x \in \mathbb{R}^n, y \in \{-1,1\}$

Initialization $w^{\circ} = 0 \in \mathbb{R}^n$

For epoch = 1 ...T:

Take a random sample (x_i, y_i) from the training set S

Consider the random sample (x_i, y_i) as a complete data set and calculate the derivative of the current SVM objective function w^{t-1} to ${}_{a}^{Caputo}D_t^{\alpha} \nabla \mathrm{J}^t(w^{t-1})$

$$ \mathrm{J}^t(w) = \frac{1}{2} w^T w + C \max(0, 1 - y_i w^T x_i) $$

Change: $w^t \leftarrow w^{t-1} - \gamma_t {}_{a}^{Caputo}D_t^{\alpha} \nabla \mathrm{J}^t(w^{t-1})$

Display the final value of w

The initial activity carried out was to collect raw data to be used, in this study ten non-linear datasets were used. The next activity is data pre-processing, eliminating data that has no value in its feature variable or on its class variable, which is referred to as the missing value problem. This is done because the SVM classifier algorithm is very sensitive to data with missing value problems. Then cross validation is carried out, namely the activity of dividing the data into training data and test data.

Then run the RBF kernel on the SVM classifier for non-linearly separable data and generate a classification model. The classification model will be optimized with fractional gradient descent of the Caputo type, followed by the implementation of the optimized model on the test data and an evaluation is carried out with the result in the form of an evaluated model. The last activity of the optimized and evaluated model is implemented on the new data.

In the optimization procedure of fractional gradient descent on the SVM classifier as follows: determine the objective function of SVM primal form without/with var penalty, determine the value of w for the SVM classifier, determine how many parameters C/slack variable (0.01; 0.0001) is used, determine the fold value for cross validation, determine the value of the learning rate (0.01; 0.001; 0.0001; 0.00001) for the calculation of SVM fractional gradient descent, construct the matrix, perform the SVM fractional gradient descent calculation process and assign the order fractional value (0.25; 0.50; 0.75), and determine the iteration scale 1000 for each given learning rate. Then a simulation of the performance of the Caputo type fractional gradient descent optimization method will be carried out on ten non-linear data.

3. Results and Discussion

In this section we have implemented the SVM fractional gradient descent classifier on ten nonlinear datasets. As shown in table 1, ten nonlinear datasets are available in UCI machine learning, consisting of the Weather dataset, an Australian weather dataset that has 5000 instances and 20 attributes. Abalone dataset is abalone species dataset with 4177 instances and 8 attributes. There are five datasets with 1000-1500 instances, including Wine, Yeast, Hepatitis, Garment and Credit datasets with

the highest number of attributes, namely 29 in the Hepatitis dataset. And three datasets with seven hundred instances such as Energy, Diabetes and Parkinson's datasets.

Table 1. Convergent time for SVM with Fractional Gradient Descent

No	Dataset	Instance	Attribute	Convergent Time	Iteration
1	Weather	5000	20	1.0273133	5
2	Abalone	4,177	8	2.9198207	17
3	Wine	1,599	12	0.01353414	10
4	Yeast	1,484	8	0.0823195	9
5	Hepatitis	1,385	29	0.064316	12
6	Garment	1,197	15	0.1379931	9
7	Credit	1,000	21	0.0902139	11
8	Energy	768	18	0.0493378	10
9	Diabetes	768	8	0.0979759	13
10	Parkinson	756	14	0.0574224	13

From table 1, it can be seen that the Abalone dataset takes the most time, then the Weather dataset is ordered, then the Garment, Diabetes, Credit, Yeast, Hepatitis, Parkinson, Energy dataset and the fastest is the Wine dataset. Diabetes dataset even though the number of instances is 768 and attribute 8 requires a long convergence time compared to the Wine dataset with the number of instances of 1599 and attribute 12.

Figure 1 shows the convergence time of the SVM classifier that has been optimized with Fractional Gradient Descent. The parameter values of the SVM classifier and the Fractional Gradient optimizer have been implemented for ten nonlinear datasets. For the slack parameter in the SVM classifier, we assign a value of 0.0001 to get the smallest error value at the convergence point. The learning rate parameter for our gradient-based optimizer gives a value of 0.01 for the most optimal results. As well as providing a value of order 0.50 for the Caputo type fractional order for optimal results

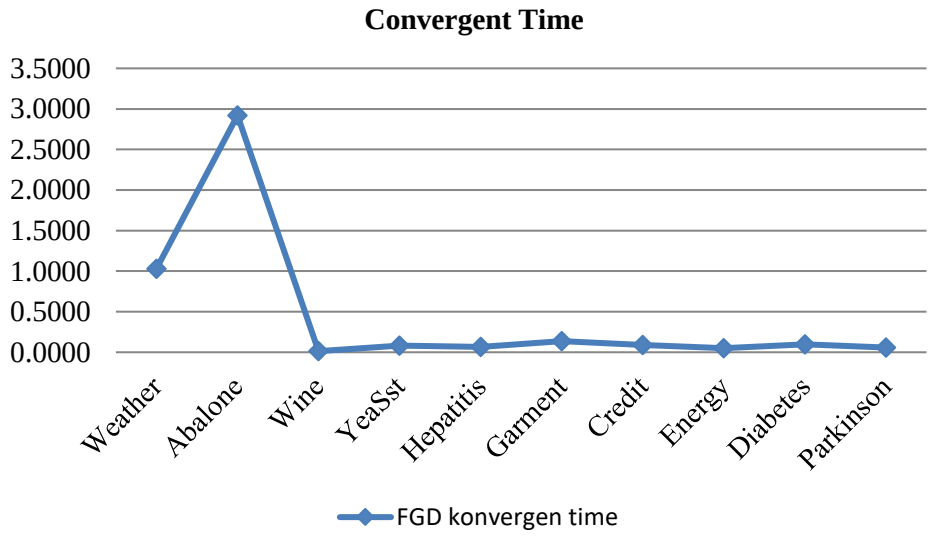

Figure 1. Convergent time SVM Fractional Gradient Descent

.

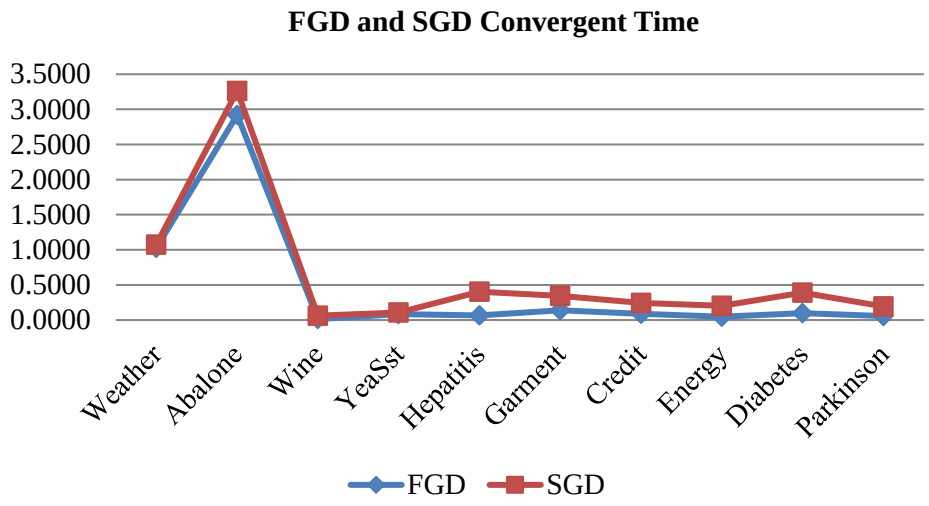

Figure 2. Comparison Convergent time

From figure 2 it can be seen that a dataset with a large number of instances takes a lot of time to reach the convergent point. The simulation results with ten nonlinear datasets show that the SVM classifier with fractional gradient optimization reaches the convergence point in a faster time than the traditional SVM classifier. The smallest convergence time occurs in the Wine dataset with 1,599 instances and 12 attributes. The largest convergence time occurs in the abalone dataset with 4,177 instances and 8 attributes. The difference in convergence time for datasets with a smaller number of instances with a large number of instances is 97%. The difference in convergent time of SVM-FGD and SVM-SGD is greater than 90% for ten nonlinear datasets. This shows that the optimization of fractional gradient descent on the SVM classifier is able to increase the convergence time.

4. Conclusion

Ten nonlinear datasets were employed in this investigation. The performance of the SVM classifier with fractional gradient optimization is evaluated using this dataset. The SVM classifier with fractional gradient optimization reaches the convergence point 90 percent faster than the classic SVM classifier, according to simulation data. The classic SVM classifier takes more iterations to reach the convergence point than the SVM classifier with fractional gradient optimization. This has an impact on the training data processing's computing speed. We plan to use an SVM classifier with fractional gradient optimization for nonlinear datasets with unbalanced classes in future research.

Acknowledgments

I would like to express my very great appreciation to the head of YPTS and ITATS as well as the head of Informatics Engineering Department for the permission given to use laboratory facilities for this research.

Referensi

[1] Wu, Hsiao and Nian, *"Using supervised machine learning on large-scale online forums to classify course-related Facebook messages in predicting learning achievement within the the personal learning environment"* - Interactive Learning Environments, Taylor & Francis, 2020

[2] Hochreiter and Schmidhuber, *"Long short-term memory"*, Neural computation, ieeexplore.ieee.org, 1997

[3] Flake and Lawrence, *"Self-organization and identification of web communities"*, Computer, ieeexplore.ieee.org, 2002

[4] Vert and Vert, *" Consistency and Convergence Rates of One-Class SVMs and Related Algorithms"*,

Journal of Machine Learning Research, jmlr.org, 2006
[5] Hsieh et al., *"LIBLINEAR: A library for large linear classification"*, the Journal of machine, jmlr.org, 2008
[6] J. Liu and X. Wu, *"New three-term conjugate gradient method for solving unconstrained optimization problems,"* ScienceAsia, 2014
[7] Bottou, *"Large-scale machine learning with stochastic gradient descent,"* in Proceedings of COMPSTAT 2010 - 19th International Conference on Computational Statistics, Keynote, Invited and Contributed Papers, 2010
[8] S. Ruder, *"Overview Optimization Gradients,"* arXiv Prepr. arXiv1609.04747, 2016
[9] Khan et al., *"Design of Momentum Fractional Stochastic Gradient Descent for Recommender Systems,"* IEEE Access, 2019
[10] S. Guo, S. Chen, and Y. Li, *"Face recognition based on convolutional neural network & support vector machine,"* in 2016 IEEE International Conference on Information and Automation, IEEE ICIA 2016, 2017
[11] Wang, Wen, et al., *"Convergence Analysis of Caputo-Type Fractional Order Complex-Valued Neural Networks,"* IEEE Access, 2017
[12] Caputo and Fabrizio, *"A new definition of fractional derivative without singular kernel,"* Prog. Fract. Differ. Appl., 2015
[13] Y. Wei, Y. Chen, S. Cheng, and Y. Wang, *"Discussion on fractional order derivatives,"* IFAC-PapersOnLine, 2017
[14] Wang, Yang, et al. al., *"Fractional-order gradient descent learning of BP neural networks with Caputo derivative,"* Neural Networks, 2017
[15] Chen and Zhao, *"An Improved Adagrad Gradient Descent Optimization Algorithm,"* in Proceedings 2018 Chinese Automation Congress, 2019.