Statistics and Computing (2024) 34:107
https://doi.org/10.1007/s11222-024-10409-5

ORIGINAL PAPER

Stochastic three-term conjugate gradient method with variance technique for non-convex learning

Chen Ouyang[1] · Chenkaixiang Lu[2] · Xiong Zhao[1] · Ruping Huang[1] · Gonglin Yuan[1] · Yiyan Jiang[3]

Received: 30 December 2023 / Accepted: 13 February 2024 / Published online: 27 March 2024

Abstract
In the training process of machine learning, the minimization of the empirical risk loss function is often used to measure the difference between the model's predicted value and the real value. Stochastic gradient descent is very popular for this type of optimization problem, but converges slowly in theoretical analysis. To solve this problem, there are already many algorithms with variance reduction techniques, such as SVRG, SAG, SAGA, etc. Some scholars apply the conjugate gradient method in traditional optimization to these algorithms, such as CGVR, SCGA, SCGN, etc., which can basically achieve linear convergence speed, but these conclusions often need to be established under some relatively strong assumptions. In traditional optimization, the conjugate gradient method often requires the use of line search techniques to achieve good experimental results. In a sense, line search embodies some properties of the conjugate methods. Taking inspiration from this, we apply the modified three-term conjugate gradient method and line search technique to machine learning. In our theoretical analysis, we obtain the same convergence rate as SCGA under weaker conditional assumptions. We also test the convergence of our algorithm using two non-convex machine learning models.

Keywords Machine learning · Empirical risk loss function minimization · Stochastic conjugate gradient · Linear convergence

1 Introduction

The core problem of algorithms in machine learning is how to find a suitable model from limited training data so that it can make accurate predictions or decisions on unknown data. To address this problem, machine learning researchers have proposed different learning criteria and optimization methods to guide the model selection and training process. The learning criterion of empirical risk minimization (ERM), which is applied to the recently popular GPT (Generative Pretrained Transformer) model. GPT is an autoregressive language model based on the Transformer structure. It can achieve excellent performance in a variety of natural language processing tasks through large-scale unsupervised pretraining and supervised fine-tuning. The training process of GPT involves the idea of empirical risk minimization Ouyang et al. (2022), that is, to improve the generalization ability of the model on the test set by minimizing the loss function of the model on the training set. Taking general linear regression as an example, assuming that the loss function is the mean square error, plus l_2 regularization, the minimization of the objective function can be written as

$$\min_{\omega,b} \frac{1}{2m}\sum_{i=1}^{m}(y_i - \omega^T x_i - b)^2 + \lambda\|\omega\|_2^2$$

✉ Gonglin Yuan
glyuan@gxu.edu.cn

Chen Ouyang
a124413760@126.com

Chenkaixiang Lu
luchen@stu.xju.edu.cn

Xiong Zhao
chao18874188217@163.com

Ruping Huang
erhuohrp@163.com

Yiyan Jiang
jiang372986@163.com

[1] School of Mathematics and Information Science, Center for Applied Mathematics of Guangxi, Guangxi University, Nanning 530004, Guangxi, China

[2] College of Civil Engineering and Architecture, Xinjiang University, Ürümqi 830047, Xinjiang, China

[3] School of Information and Communication, Guilin University of Electronic Technology, Guilin 541214, Guangxi, China

where $w \in \Re^L$ is the weight vector of the model, b is the intercept, y_i is the observed value of sample i, and x_i is the feature vector of sample i. n is the total number of samples and λ is the regularization hyperparameter. We use $F_i : \Re^L \to \Re$ to denote the objective function for the i-th sample, and then the above can be written in a more general form:

$$\min_{\omega} F(\omega) = \frac{\sum_{i=1}^{m} F_i(\omega)}{m} \tag{1.1}$$

The full gradient descent algorithm Cauchy (1847) is a classic algorithm to solve the above problems. The basic idea is that each time the parameters are updated, the gradient information on the entire training set is used to advance a certain step in the opposite direction of the gradient, thereby gradually reducing the value of the loss function until it converges to a local minimum or global min. However, when the training set is large, calculating the gradient will be very time-consuming, and the same gradient must be calculated repeatedly every time the parameters are updated, resulting in inefficiency. In addition, the full gradient descent algorithm is also sensitive to the choice of learning rate. If the learning rate is too large or too small, it will affect the convergence speed and effect. In order to overcome the shortcomings of the full gradient descent algorithm, many improved methods have appeared: Bottou (2010); Bottou et al. (2018); Robbins and Monro (1951); Goodfellow et al. (2016). At the same time, in order to increase the convergence rate of SGD algorithms, Le Roux Schmidt et al. (2017) proposed an SGD method with variance technology, and based on this work, more gradient methods with variance technology such as CGVR (Jin et al. (2018)), SCGN (Yuan et al. (2021a)), SCGA (Kou and Yang (2022)), and their common feature is that they all use the stochastic conjugate gradient method. Among them, CGVR and SCGA are both hybrid conjugate gradient methods, and both achieve linear convergence rates under strong convex conditions. In addition to this, there are also adaptive methods that are popular in machine learning such as: AdaGrad (Lydia and Francis (2019)), AdaDelta (Zeiler (2012)), Adam (Kingma and Ba (2014)). Due to their adaptive step size and relatively robust selection of hyper-parameters, they perform well on many problems even without fine-tuning hyper-parameters. With the increasing size of data in machine learning problems, and the good performance of traditional conjugate gradient methods in dealing with large-scale equations, we have reason to believe that the stochastic conjugate gradient method can be better applied in machine learning.

1.1 Conjugate gradient method

Considering the traditional unconstrained optimization problem:

$$\min\{f(x) \mid x \in \Re^L\}, \tag{1.2}$$

where $f : \Re^L \to \Re$. The conjugate gradient method is a important algorithm to solve the problem (1.2), which is a method between the steepest descent method and the Newton method. It overcomes the slow convergence of the steepest descent method and avoids the disadvantage of the Newton method that needs to calculate the Hesse matrix. The CG method, which is defined by

$$d_{s+1} = \begin{cases} -g_{s+1} + \beta_s d_s, & \text{if } s \geq 0, \\ -g_{s+1}, & \text{if } s = 0, \end{cases} \tag{1.3}$$

where $g_{s+1} = \nabla f(x_{s+1})$ is the gradient of $f(x)$ at x_{s+1}, and $\beta_s \in \Re$ is a scalar which has four classic CG formulas with

$$\beta_s^{PRP} = \frac{g_{s+1}^T y_{s+1}}{\|g_s\|^2}, \ [22, 23, 27]$$

$$\beta_s^{FR} = \frac{g_{s+1}^T g_{s+1}}{\|g_s\|^2}, \ [20]$$

$$\beta_s^{HZ} = \frac{g_{s+1}^T y_{s+1}(d_s^T y_{s+1} - 2\|y_{s+1}\|^2(g_{s+1}^T d_s))}{y_{s+1}^T d_s}, \ [21]$$

$$\beta_s^{DY} = \frac{g_{s+1}^T g_{s+1}}{y_{s+1}^T d_s}, \ [25],$$

where $\|\cdot\|$ denote the Euclidean norm and $y_{k+1} = g_{s+1} - g_s$. For better theoretical or numerical results, many scholars have revised these classical directions Yuan et al. (2022a, 2021b, a, 2022b, 2019); Wang et al. (2022). Among them, Yuan et al. (2019) obtained the global convergence of PRP through a modified wolf line search. Recently, the adaptive conjugate gradient method proposed by Wang and Ye (2023) has a better performance in training neural networks for image processing. CGVR is a hybrid of FR and PRP methods on the basis of SVRG (Johnson and Zhang (2013)), and SCGA is a similar work on SAGA (Defazio and Bach (2014)). Although both of them exhibit faster convergence rates experimentally, both require strong theoretical assumptions in convergence analysis, which is worthy of our further study. Jiang et al. (2023) weakens the hypothesis by restarting the coefficients. In order to weaken the condition of the assumption, it prompts us to think about the direction of descent and the step size. Inspired by Yuan, we consider using line search to get some better theoretical properties. As we all know, when we require the step size to satisfy the strong wolf step size condition Wolfe (1969), we can avoid the case where the direction is not the descending direction,

$$\begin{aligned} &f(x_s + \alpha_s d_s) \leq f(x_s) + \eta\alpha_s \nabla f(x_s)^T d_s, \\ &|\nabla f(x_s + \alpha_s d_s)^T d_s| \leq -\sigma \nabla f(x_s)^T d_s \end{aligned} \tag{1.4}$$

where $0 < \eta < \sigma < 1$. Due to its strong properties and numerical effects, some scholars use the strong wolfe condition in stochastic optimization problems (Kou and Yang (2022); Jin et al. (2018)).

2 Inspiration and algorithm

Inspired by many scholars and the conjugate gradient and line search for continuous optimization in the introduction, we want to apply the new conjugate gradient method and line search to stochastic optimization problems. Here we first review two different gradient estimates in stochastic methods. In stochastic optimization algorithms, using the mini-batch sampling method can improve computational speed, avoid redundant samples, and accelerate convergence. This method achieves acceleration by processing only a small portion of the data per iteration, while ensuring accuracy of results. It significantly reduces computational costs and is more easily applicable to large-scale datasets. The gradient estimation expression of the mini-batch SAGA algorithm is as follows::

$$G_s = \nabla F_{C_s}(\omega_s) - \frac{\sum_{i\in C_s} \nabla F_i(\omega_{[s-1]})}{|C_s|} + \frac{\sum_{i=1}^{m} \nabla F_i(\omega_{[s-1]})}{m} \tag{2.1}$$

where $\omega_{[s]}$ represents the latest iterate at which ∇F_i was evaluated, C_s represents the s-th small batch and $|C_s|$ represents the size of the batch. $\nabla F_i(\omega_{[s]})$ is the gradient of the i-th sample at iterate $\omega_{[i]}$. Taking the expectation from the above equation can be seen that it is an unbiased estimate. The first three-term CG formula is presented by Zhang et al. (2007) for continuous optimization problems (1.2):

$$d_{s+1} = \begin{cases} -g_{s+1} + \frac{g_{s+1}^T y_{s+1} d_s - g_{s+1}^T d_s y_{s+1}}{\|g_s\|^2}, & \text{if } s \geq 1 \\ -g_{s+1}, & \text{if } s = 0, \end{cases} \tag{2.2}$$

we write in a more general form:

$$d_{s+1} = -g_{s+1} + \beta_{s+1} d_s - \theta_{s+1} y_{s+1}$$

where $y_{s+1} = g_{s+1} - g_s$, β_{s+1} is a parameter of the standard PRP conjugate gradient method, θ_{s+1} is a parameter of the three-term CG method. β_{s+1} and θ_{s+1} are calculated as follows:

$$\beta_{s+1} = \frac{g_{s+1}^T y_{s+1}}{\|g_s\|^2}, \quad \theta_{s+1} = \frac{g_{s+1}^T d_s}{\|g_s\|^2} \tag{2.3}$$

Some studies on the three-term conjugate gradient method have shown that it has the good property. In Kim et al. (2023), he applied the three-term conjugate gradient method to the artificial neural network, which is comprehensively compared with SGD, Adam, AMSGrad Reddi et al. (2019) and AdaBelief Zhuang et al. (2020) methods and is competitive, but it doesn't do much theoretical analysis. Kou and Yang (2022) tried to add the conjugate gradient method to SAGA and got SCGA. Yang (2022) combining mini-batch SARAH (Nguyen et al. (2017)) with FR conjugate gradient methods named CG-SARAH-SO. Huang et al. (2023) combined the modified PRP with SARAH to propose BSCG. Inspired by Yang, Kou and Kim, can we take advantage of the property of the three-term conjugate gradient to obtain the convergence rate estimated in other gradients? we try to use the excellent properties of conjugate gradient directions, remove some assumptions, and adopt a new direction inspired:

$$d_{s+1} = \begin{cases} -g_{s+1} + \frac{y_{s+1}^T g_{s+1} d_s - d_s^T g_{s+1} y_{s+1}}{\mu_1 \|y_{s+1}\| \|d_s\| + \mu_2 \|y_{s+1}\| \|g_{s+1}\| + \|g_s\|^2}, & \text{if } s \geq 1 \\ -g_{s+1}, & \text{if } s = 0, \end{cases} \tag{2.4}$$

we try to apply direction (2.4) to the stochastic optimization problem:

$$P_{s+1} = \begin{cases} -G_{s+1} + B_{s+1} P_s - O_{s+1} Y_{s+1}, & \text{if } s \geq 1 \\ -G_{s+1}, & \text{if } s = 0, \end{cases} \tag{2.5}$$

where P_{s+1} represents the descending direction of the sample iteration and $Y_{s+1} = G_{s+1} - G_s$. B_{s+1} and O_{s+1} are calculated as follows:

$$B_{s+1} = \frac{G_{s+1}^T Y_{s+1}}{\mu_1 \|Y_{s+1}\| \|P_s\| + \mu_2 \|Y_{s+1}\| \|G_{s+1}\| + \|G_s\|^2}, \tag{2.6}$$

$$O_{s+1} = \frac{G_{s+1}^T P_s}{\mu_1 \|Y_{s+1}\| \|P_s\| + \mu_2 \|Y_{s+1}\| \|G_{s+1}\| + \|G_s\|^2} \tag{2.7}$$

Dai (2002) proposes two Armjio-style line search methods, one of which is as follows:

$$\begin{aligned} f(x_{s+1}) &\leq f(x_s) + \eta \alpha_s g_s^T d_s, \\ g_{s+1}^T d_{s+1} &\leq -\sigma_2 \|g_{s+1}\|^2 \end{aligned} \tag{2.8}$$

given constants $t \in (0,1), \eta > 0, \sigma_2 \in (0,1]$, $\alpha_s = max\{t, t^2, ...\}$. In fact, the second search line of this line includes sufficient descent, but because it uses the direction when $s = s+1$, it leads to a large amount of calculation when looking for the step size. In order to make the step size more acceptable, we are in a correction item has also been added to the Armjio condition. Inspired by Yuan et al. (2019) of modifying Wolfe's line search criterion, we modify Dai's line search. In order to get an appropriate step size α_k, we

design a modifed inexact line search:

$$f(x_{s+1}) \leq f(x_s)+max\{\eta_1\alpha_s g_s^T d_s, -\eta_2\alpha_s^2\|g_s\|^2\}, \quad g_{s+1}^T d_{s+1}+\sigma_1\|d_{s+1}\|^2 \leq -\sigma_2\|g_{s+1}\|^2 \tag{2.9}$$

where η_1 and η_2 are constants in (0,1) and $\sigma_2+\sigma_1 > 1, \sigma_1 \leq 1, \sigma_2 > 0$, when σ_1 is small enough, it approximates the second criterion of (2.8). It can lead to some useful conclusions with the nature of the direction when discussing the convergence of the algorithm later. We write (2.9) in the form of a random line search and apply it to a stochastic optimization problem to find the step size:

$$F_C(x_{s+1}) \leq F_C(x_s) + max\{\eta_1\alpha_s G_s^T P_s, -\eta_2\alpha_s^2\|G_s\|^2\}, \quad G_{s+1}^T P_{s+1} + \sigma_1\|P_{s+1}\|^2 \leq -\sigma_2\|G_{s+1}\|^2 \tag{2.10}$$

Direction (2.1) calculated by the conjugate gradient method and modified line search (2.10), we give the algorithm SATCG. Due to the properties of sufficient descent and trust region, we have reasons to believe that SATCG possesses better theoretical properties compared to SCGA and CGVR.

The remainder of this paper is organized as follows. In Section 1, we review the SGD algorithm, gradient estimation methods with variance reduction techniques, and conjugate gradient methods. In Section 2, inspired by the SCGA and three conjugate gradient methods, we propose the SATCG algorithm. In Section 3, we provide a theoretical analysis of SATCG. In Section 4, we analyze the experimental performance of SATCG in two machine learning models.

2.1 Contribution

◇ We provide a convergence analysis of SATCG in the non-convex condition, and compared to SCGA and CGVR, SATCG exhibits superior theoretical properties, achieving linear convergence rate with fewer assumptions.

◇ We analyze the performance of SATCG in two machine learning models and find that it exhibits better numerical performance than traditional SGD methods on large-scale datasets. We also compare it to adaptive algorithms. Additionally, we investigate the performance of SATCG and SCGA on small-sized datasets. Overall, SATCG demonstrates superior convergence properties compared to traditional SGD algorithms and remains competitive with SCGA, while providing more stable numerical performance.

3 Features and convergence of SATCG

Assumption 3.1 For all of the individual function $\nabla F_i(\omega)$ is Lipschitz smooth. From the properties of Lipschitz, we can obtain the following conclusion:

$$F(\omega) \leq F(v)+ < \nabla F(v), \omega - v > +\frac{L}{2}\|\omega - v\|^2 \tag{3.1}$$

Assumption 3.2 The Polyak-ojasiewicz (PL) condition holds for some $\Omega > 0$.

$$F(\omega) - F(\omega_*) \leq 2\Omega\|\nabla F(\omega)\|^2$$

where $F(\omega_*)$ is the lower bound on the function F. The PL condition does not contain any convexity, an example in the article in Karimi et al. (2016): $f(x) = x^2 + 3sin^2 x$, when $\Omega = 8$, the PL condition is satisfied, but the function is non-convex.

Theorem 3.1 *We can derive the following two properties based on Direction 1:*

$$G_s^T P_s = -\|G_s\|^2 \tag{3.2}$$

and

$$\|P_s\| \leq (1 + \frac{2}{\mu_1})\|G_s\| \tag{3.3}$$

Algorithm 1 SATCG

1: Choose an initial iterate $\omega_{[0]} \in \Re^L$,and step size α_0
2: **for** i=1,2,...,m **do**
3: Compute $\nabla F_i(\omega_0)$
4: Store $\nabla F_i(\omega_{[0]}) \leftarrow \nabla F_i(\omega_0)$
5: **end for**
6: Set the initial stochastic gradient $G_0 = \frac{1}{m}\sum_{i=1}^{m} \nabla F_i(\omega_0)$
7: Set the initial direction $P_0 = -G_0$
8: **for** s=1,2,... **do**
9: Find the step size α_{s-1} satisfying (2.10).
10: Update iterate $\omega_s \leftarrow \omega_{s-1} + \alpha_{s-1}P_{s-1}$.
11: Choose mini-batch $C \subset \{1,...,n\}$ of size b uniformly random, compute (2.1)
12: Set $G_s \leftarrow \nabla F_{C_s}(\omega_s) - \mu_{C_s} + \mu_{s-1}$
13: Compute B_s, O_s by

$$B_s = \frac{G_s^T Y_s}{\mu_1\|Y_s\|\|P_{s-1}\| + \mu_2\|Y_s\|\|G_s\| + \|G_{s-1}\|^2},$$

$$O_s = \frac{G_s^T P_{s-1}}{\mu_1\|Y_s\|\|P_{s-1}\| + \mu_2\|Y_s\|\|G_s\| + \|G_{s-1}\|^2}$$

14: Determine $P_s \leftarrow -G_s + B_s P_{s-1} - O_s Y_s$
15: Update $\nabla F_j(\omega_{[s]}) \leftarrow \nabla F_j(\omega_s), \forall \in C_s$, while other entries of the stored full gradient remain unchanged.
16: Update $\mu_s \leftarrow \frac{1}{m}\sum_{i=1}^{m} \nabla F_i(\omega_{[s]})$
17: **end for**

Proof When s=1, (3.2) and (3.3) obviously hold. For $s > 2$, by (2.4) we have

$$\begin{aligned} G_s^T P_s &= G_s^T[-G_s + \frac{Y_s^T G_s P_{s-1} - P_{s-1}^T G_s Y_s}{\mu_1\|Y_s\|\|P_{s-1}\| + \mu_2\|Y_s\|\|G_s\| + \|G_{s-1}\|^2}] \\ &= -\|G_s\|^2 + \frac{Y_s^T G_s P_{s-1}^T G_s - P_{s-1}^T G_s Y_s^T G_s}{\mu_1\|Y_s\|\|P_{s-1}\| + \mu_2\|Y_s\|\|G_s\| + \|G_{s-1}\|^2} \\ &= -\|G_s\|^2 \end{aligned}$$

We can get: $G_s^T P_s = -\|G_s\|^2 \geq -\|G_s\|\|P_s\|$, then $\|P_s\| \geq \|G_s\|$. By (2.4) we have

$$\begin{aligned} \|P_s\| &= \| - G_s + \frac{Y_s^T G_s P_{s-1} - P_{s-1}^T G_s Y_s}{\mu_1\|Y_s\|\|P_{s-1}\| + \mu_2\|Y_s\|\|G_s\| + \|G_{s-1}\|^2}\| \\ &\leq \|G_s\| + \frac{2\|Y_s\|\|G_s\|\|P_{s-1}\|}{\mu_1\|Y_s\|\|P_{s-1}\| + \mu_2\|Y_s\|\|G_s\| + \|G_{s-1}\|^2} \\ &\leq \|G_s\| + \frac{2\|Y_s\|\|G_s\|\|P_{s-1}\|}{\mu_1\|Y_s\|\|P_{s-1}\| + \mu_2\|Y_s\|\|G_s\| + \|G_{s-1}\|^2} \\ &\leq (1 + \frac{2}{\mu_1})\|G_s\| \end{aligned}$$

In summary, (3.2) and (3.3) are established. The theorem is still satisfied when the direction is (2.4) and the line search is (2.9).

Lemma 3.1 *Suppose that x_1 is a starting point that satisfies satisfy the gradient L-smooth. Consider the descending direction to satisfy (2.4), where the stepsize α_s is determined through line search (2.9). In this case, for every s, line search will compute a positive stepsize $\alpha_s > 0$ and generate a descent direction d_{s+1}. Furthermore, it can be shown that:*

$$\alpha_s \geq min\{1, c_1\}, c_1 = \frac{2t(1-\eta_1)}{L}$$

So we can say that there will be an upper and lower bound on the step size α_s that satisfies $\alpha_s \in [\alpha_1, 1]$, $0 < \alpha_1 < 1$.

Proof Since $d_1 = -g_1$, d_1 is a descent direaction.

For any $\bar{\alpha_s}$, define $x_{s+1} = x_s + \bar{\alpha_s} d_s$, similar to proofs in Dai (2002). As theorem 3.1 shows $g_s^T d_s = -\|g_s\|^2$, then

$$\begin{aligned} -\|g_{s+1}\|^2 + \sigma_1\|d_{s+1}\|^2 \leq -\sigma_2\|g_{s+1}\|^2 \leq -\frac{\sigma_2}{(1+\frac{2}{\mu_1})^2}\|d_{s+1}\|^2 \\ (\sigma_1 + \frac{\sigma_2}{(1+\frac{2}{\mu_1})^2})\|d_{s+1}\|^2 \leq \|g_{s+1}\|^2 \end{aligned}$$

Because $\sigma_2 + \sigma_1 > 1$ and $\|P_s\| \leq (1 + \frac{2}{\mu_1})\|G_s\|$ holds. The above equation clearly exists. From the properties of Lipschitz, then we have that

$$\begin{aligned} &f(x_{s+1}) - f(x_s) \leq \eta_1\bar{\alpha_s} g_s^T d_s \\ &\leq max\{\eta_1\bar{\alpha_s} g_s^T d_s, -\eta_2\bar{\alpha_s}\|g_s\|^2\}, \\ &\quad for\ all\ \bar{\alpha_s} \in (0, \frac{2t(1-\eta_1)}{L}\frac{|g_s^T d_s|}{\|d_s\|^2}) \end{aligned}$$

Because the $g_s^T d_s = -\|g_s\|^2$, by theorem 3.1, we can derive the $\bar{\alpha_s} \in (0, \frac{2(1-\eta_1)}{L})$. There due to line search determines a positive stepsize $\bar{\alpha_s} > 0$ and further, above holds with constant $c_1 = \frac{2t(1-\eta_1)}{L}$.

Theorem 3.2 *Assuming that the step size satisfies condition 1 and is in the direction of 2.5, then the gradient G_s satisfy*

$$\frac{\|G_{s+1}\|}{\|G_s\|} \leq \frac{2\mu_1 + 4}{\mu_1\mu_2(1 - \frac{1-\sigma_1}{\sigma_2})} = \beta$$

It is evident that when $\frac{2\mu_1+4}{\mu_1\mu_2} < \frac{\sigma_2-\sigma_1-1}{\sigma_2}$, the inequality $\beta = \frac{\|g_{s+1}\|}{\|g_s\|} < 1$ holds true.

Proof Multiply P_{s+1} at both ends of the direction (2.4)

$$\begin{aligned} \|P_{s+1}\|^2 &= -P_{s+1}^T G_{s+1} \\ &\quad + \frac{Y_{s+1}^T G_{s+1} P_{s+1}^T P_s - P_s^T G_{s+1} P_{s+1}^T Y_{s+1}}{\mu_1\|Y_{s+1}\|\|P_s\| + \mu_2\|Y_{s+1}\|\|G_{s+1}\| + \|G_s\|^2} \\ &\leq \|G_{s+1}\|^2 + \frac{2\|Y_{s+1}\|\|G_{s+1}\|\|P_{s+1}\|\|P_s\|}{\mu_1\|Y_{s+1}\|\|P_s\| + \mu_2\|Y_{s+1}\|\|G_{s+1}\| + \|G_s\|^2} \\ &\leq \frac{1-\sigma_1}{\sigma_2}\|P_{s+1}\|^2 + \frac{2}{\mu_2}\|P_{s+1}\|\|P_s\| \end{aligned}$$

The inequality in the last line is due to the second line of the search (2.10):

$$\begin{aligned} -\|P_{s+1}\|^2 + \sigma_1\|P_{s+1}\|^2 &\leq G_{s+1}^T P_{s+1} + \sigma_1\|P_{s+1}\|^2 \\ &\leq -\sigma_2\|G_{s+1}\|^2 \end{aligned}$$

By (3.2) and (3.3), when the $\|P_{s+1}\| \neq 0$, we get

$$\begin{aligned} (1 - \frac{1-\sigma_1}{\sigma_2})\|G_{s+1}\| &\leq (1 - \frac{1-\sigma_1}{\sigma_2})\|P_{s+1}\| \\ &\leq \frac{2}{\mu_2}(1 + \frac{2}{\mu_1})\|G_s\| \end{aligned}$$

then we have

$$\frac{\|G_{s+1}\|}{\|G_s\|} \leq \frac{2\mu_1 + 4}{\mu_1\mu_2(1 - \frac{1-\sigma_1}{\sigma_2})}$$

Lemma 3.2 *Let ω^* be the unique minimizer of F. Taking expectation with respect to C_s of $\|G_0\|^2$, we obtain*

$$E[\|G_0\|^2] \leq 2L[F(\omega_0) - F(\omega_*)]$$

For details, see Lemma 4 of the Kou and Yang (2022).

Theorem 3.3 *Suppose that Assumptions 3.1, and Theorem 3.2 hold. Let ω_* be the unique minimizer of F. Then, for all $s \geq 0$, we have taking expectation in this relation conditioned*

on C_s. From Lemma 3, it can be known that the step size searched by line search (2.9) has upper and lower bounds, so the step size of (2.10) also has upper and lower bounds. Assuming that we choose $\alpha_1 < 2\Omega(1-\beta^2)$, we can get the linear convergence rate of the algorithm.

Proof By $\omega_{s+1} = \omega_s + \alpha_s P_s$, the inequality (3.1) can be written as

$$F_{C_s}(\omega_{s+1}) - F_{C_s}(\omega_s) \le \alpha_s G_s^T d_s + \frac{1}{2}\alpha_s^2 L \|P_s\|^2$$

On C_s taking expectations on both sides of inequality,

$$\begin{aligned}
&E[F(\omega_{s+1})] - E[F(\omega_s)] \\
&\le E[\alpha_s G_s^T P_s] + \frac{1}{2}\alpha_s^2 L E[\|P_s\|^2] \\
&\le -\alpha_s E[\|G_s\|^2] + \frac{1}{2}\alpha_s^2 L(1+\frac{2}{\mu_1})^2 E[\|G_s\|^2] \\
&\le -\alpha_1 E[\|G_s\|^2] \\
&+ [\frac{L\alpha_s^2}{2}(1+\frac{2}{\mu_1})^2] E[\|G_s\|^2] \\
&\le -\alpha_1 \|E[G_s]\|^2 + [\frac{L\alpha_s^2}{2}(1+\frac{2}{\mu_1})^2]\beta^{2s} E[\|G_0\|^2] \\
&\le -\alpha_1 \|\nabla F(\omega_s)\|^2 \\
&+ [\frac{L}{2}(1+\frac{2}{\mu_1})^2]\beta^{2s} 2L E[F(\omega_0) - F(\omega_*)]
\end{aligned}$$

The second inequality uses (3.3), the third inequality can be obtained from Lemma 3.1, and the fourth inequality uses Theorem 3.2. The last line can be obtained by Lemma 3.2 and Lemma 3.3. Adding the expectation of $F(\omega_*)$ to both sides of the inequality, we get

$$\begin{aligned}
&E[F(\omega_{s+1})] - E[F(\omega_*)] \\
&\le (1 - \frac{\alpha_1}{2\Omega})(E[F(\omega_s)] - E[F(\omega_*)]) \\
&\quad + [L^2(1+\frac{2}{\mu_1})^2]\beta^{2s} E[F(\omega_0) - F(\omega_*)]
\end{aligned}$$

Then, we define

$$\begin{aligned}
\xi &= 1 - \frac{\alpha_1}{2\Omega} \\
T(i) &= L^2(1+\frac{2}{\mu_1})^2\beta^{2i} \\
T &= 1 + L^2(1+\frac{2}{\mu_1})^2 \frac{1}{\xi - \beta^2} \\
X_{s+1} &= E[F(\omega_{s+1})] - E[F(\omega_*)]
\end{aligned}$$

The above formula can be expressed as:

$$\begin{aligned}
X_{s+1} &\le (1 - \frac{\alpha_1}{2\Omega})X_s + T(s)X_0 \\
&\le (1 - \frac{\alpha_1}{2\Omega})^2 X_{s-1} + T(s-1)X_0 + T(s)X_0
\end{aligned} \tag{3.4}$$

We scale the right side of (3.4) to $s = 0$ has the following formula:

$$\begin{aligned}
X_{s+1} &\le [\xi^{s+1} + \sum_{i=0}^{s} \xi^{s-i} T(i)]X_0 \\
&\sum_{i=0}^{s} \xi^{s-i} T(i) = \xi^s L^2(1+\frac{2}{\mu_1})^2 \sum_{i=0}^{s}(\frac{\beta^2}{\xi})^i \\
&= \xi^s L^2(1+\frac{2}{\mu_1})^2 \frac{1 - (\frac{\beta^2}{\xi})^{s+1}}{1 - \frac{\beta^2}{\xi}}
\end{aligned}$$

Through the above conclusions, we can get the linear convergence rate of the algorithm 1

$$\begin{aligned}
&E[F(\omega_{s+1})] - E[F(\omega_*)] \le [\xi^{s+1} \\
&+ \xi^s L^2(1+\frac{2}{\mu_1})^2 \frac{1}{1 - \frac{\beta^2}{\xi}}] E[F(\omega_0) - F(\omega_*)] \\
&\le \xi^{s+1}[1 + L^2(1+\frac{2}{\mu_1})^2 \frac{1}{\xi - \beta^2}] E[F(\omega_0) - F(\omega_*)] \\
&\le \xi^{s+1} T E[F(\omega_0) - F(\omega_*)]
\end{aligned}$$

4 Applications of algorithm in machine learning models

We used the following two models to evaluate our algorithm, mainly for the following two reasons:

1. These two models are machine learning models using non-convex sigmoid loss function, which can better adapt to the distribution of data and Noise, improve the robustness and generalization ability of the model. All the codes are written in MATLAB 2018a on a PC with a 12th Gen Intel(R) Core(TM) i7-12650 H 2.30 GHz and 16 GB of memory.

2. These two models represent two different regularization strategies: Nonconvex regularized ERM model uses non-convex regularization terms, such as ℓ_0 norm or ℓ_p norm ($0 < p < 1$), To enhance the sparsity of the model; the Non-convex SVM model uses the ℓ_2 norm as a regularization term to control the complexity of the model. Both strategies have their own advantages and disadvantages, and we hope to analyze the performance of our algorithm under different regularization settings by comparing their performance (Tables 1, 2).

Table 1 Dataset

Data set	Training samples	Dimension
Adult	32562	123
Covtype	581012	54
Mnist	60000	784
Ijcnn	49990	22
A9a	32561	123
w8a	49749	300
Diabetes	768	8
Fourclass	862	2
German.numer	1000	24
Inosphere	351	34
Splice	1000	60
Sonar	208	60

All of the above datasets have been scaled to the $[-1, 1]$ range via max-min through the preprocessing phase.

Test model 1

Nonconvex SVM model with a sigmoid loss function

$$min\ \frac{1}{n}\sum_{i=1}^{n} F_i(\omega) + \sigma\|\omega\|^2$$

where $F_i(\omega) = 1 - tanh(v_i < \omega, u_i >), u \in \Re$ and $v \in \{-1, 1\}$ represent the feature vector and corresponding label respectively.

Test model 2

Nonconvex regularized ERM model with a nonconvex sigmoid loss function

$$min\ \frac{1}{n}\sum_{i=1}^{n} F_i(\omega) + \frac{\sigma}{2}\|\omega\|^2$$

where $F_i(\omega) = \frac{1}{[1+exp(b_i a_i^T \omega)]}$. Binary classification problem is a common type of problem in machine learning, which requires us to divide the data into two categories, such as distinguishing spam and normal emails, or distinguishing whether a sonar signal is a rock or a metal.

Algorithm comparison

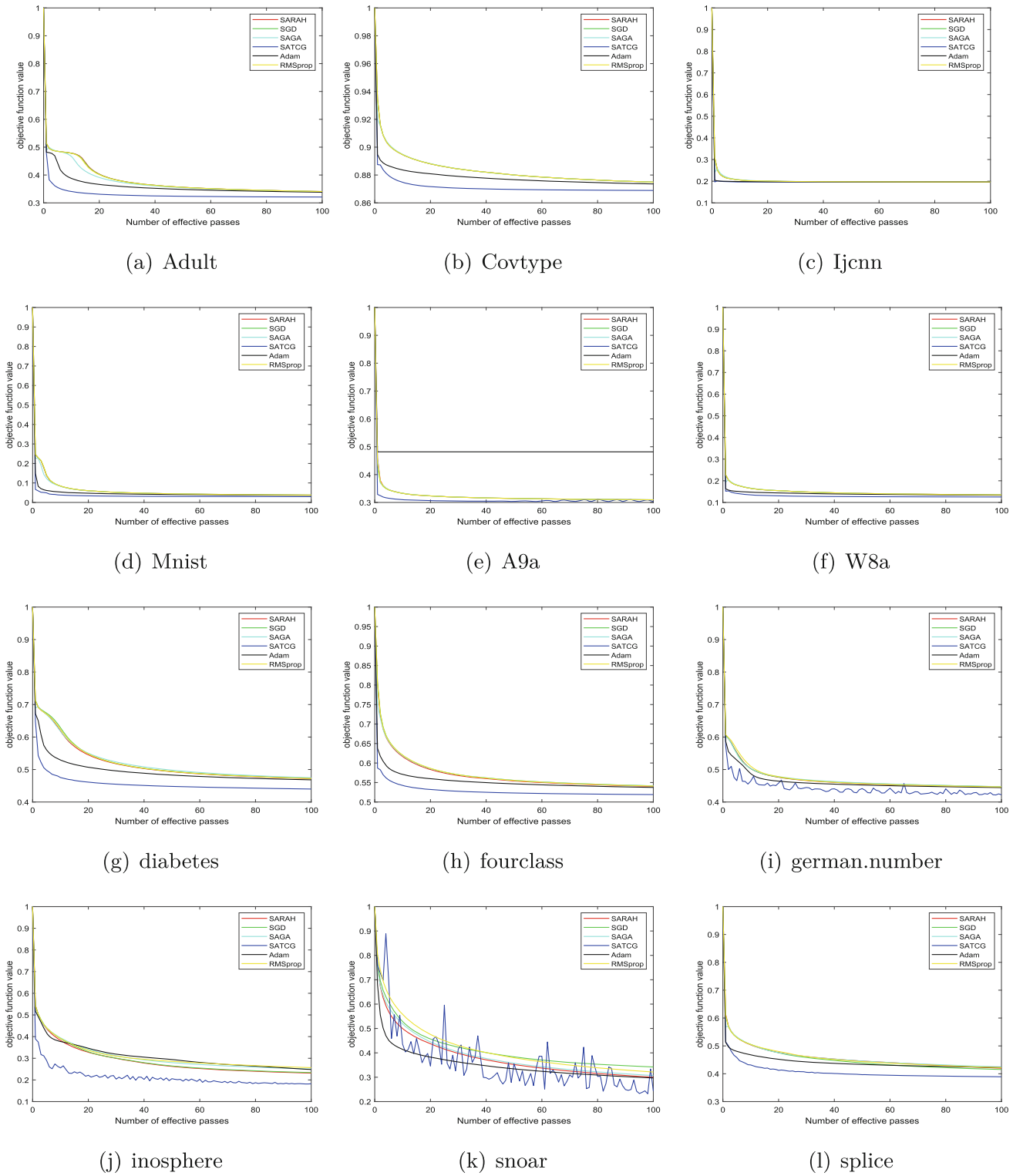
In both model tests, SGD, SAGA, SARAH, Adam, and Rmsprop all use an initial step size of 0.1, while Adam has momentum parameters of 0.9 and 0.999, and Rmsprop has a momentum parameter of 0.99, which are commonly used settings. The results are presented in Figs. 1 and 2, where we compare the convergence of SATCG and SGD class algorithms with adaptive algorithms on 8 different datasets. It is evident that SATCG generally achieves faster convergence on all datasets and models.

In Figs. 3 and 4, we compare a stochastic conjugate gradient method known as SCGA. Kou's study also compares SCGA with CGVR in several small-scale datasets. In addition, we compare the convergence of SATCG and SCGA in six smaller datasets. From the experimental results of Comparative Model 1, we observe that SATCG and SCGA perform similarly in the diabetes and fourclass datasets, but SATCG exhibits better descent and stability in the remaining four datasets. But in Model 2, SCGA has a better performance.

Figures 5 and 6 show the performance of SATCG with different regularization coefficients on six small-scale data sets. It can be seen that SATCG has better experimental results and faster decline when the regularization coefficient is smaller.

Table 2 Parmeter value

Parmeter	Value	Description
α	0.1	The initial step size in Adam, Rmsprop, SAGA, SGD, SARAH
b	$[\sqrt{m}]$	Mini batch quantity(the square root of the dataset dimension is rounded down)
μ_1	2	Direction correction factor
μ_2	5	Direction correction factor
σ_1	0.9	Line search second criterion coefficients
σ_2	0.9	Line search second criterion coefficients
η_1	0.003	Armjio criterion coefficient
η_2	0.003	Armjio criterion coefficient
λ	10^{-5}	Regularization coefficient of the loss function



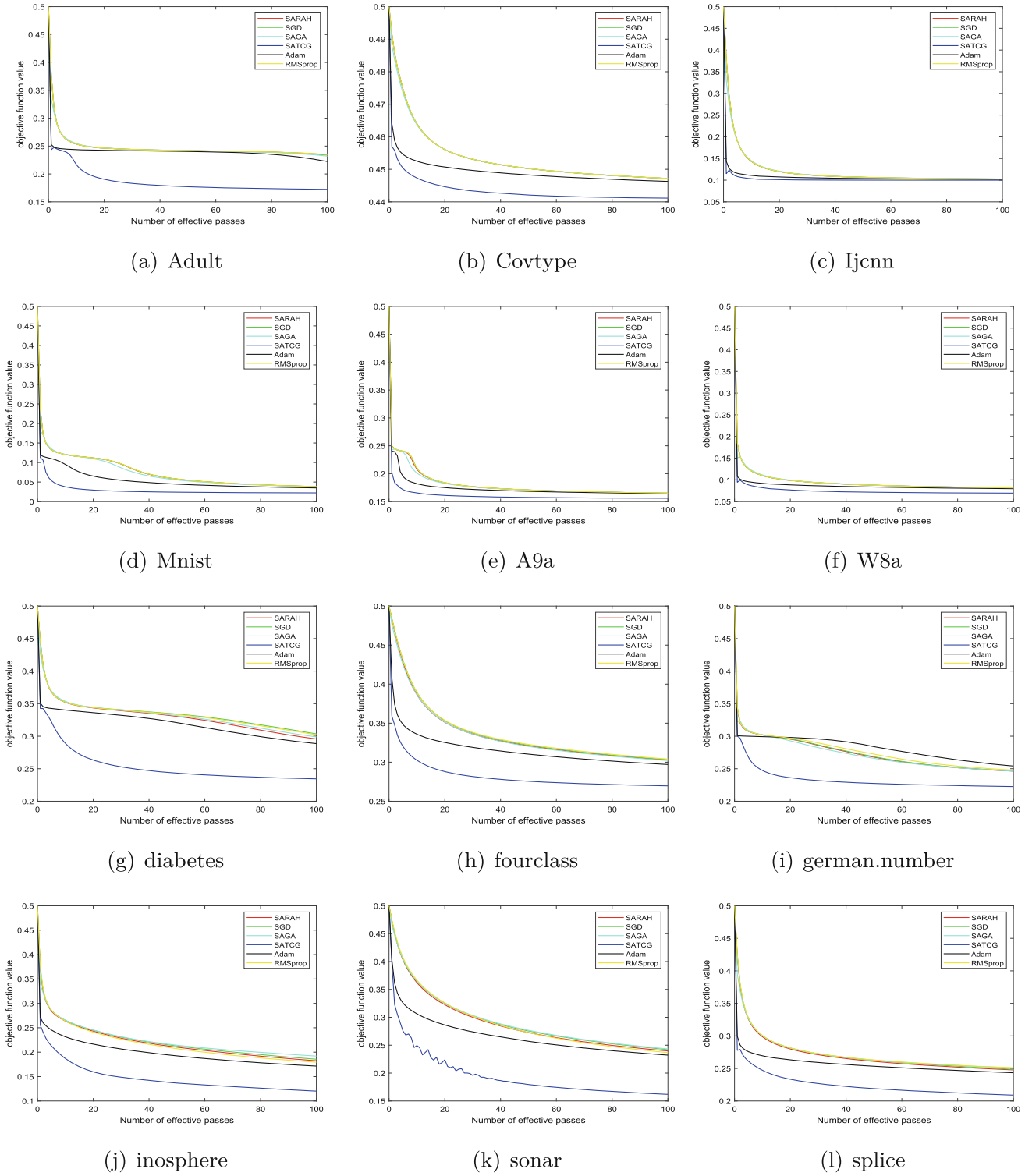
(a) Adult (b) Covtype (c) Ijcnn

(d) Mnist (e) A9a (f) W8a

(g) diabetes (h) fourclass (i) german.number

(j) inosphere (k) snoar (l) splice

Fig. 1 Comparison of SATCG with several classes of algorithms on 12 datasets adult, covtype, ijcnn, mnist, a9a, w8a, diabetes, fourclass, german.number, inosphere, snoar, splice for Model 1.(The x-axis represents numbers of iterations. The y-axis represents loss function values.)



(a) Adult (b) Covtype (c) Ijcnn

(d) Mnist (e) A9a (f) W8a

(g) diabetes (h) fourclass (i) german.number

(j) inosphere (k) sonar (l) splice

Fig. 2 Comparison of SATCG with several classes of algorithms on 12 datasets adult, covtype, ijcnn, mnist, a9a, w8a, diabetes, fourclass, german.number, inosphere, snoar, splice for Model 2.(The x-axis represents numbers of iterations. The y-axis represents loss function values.)

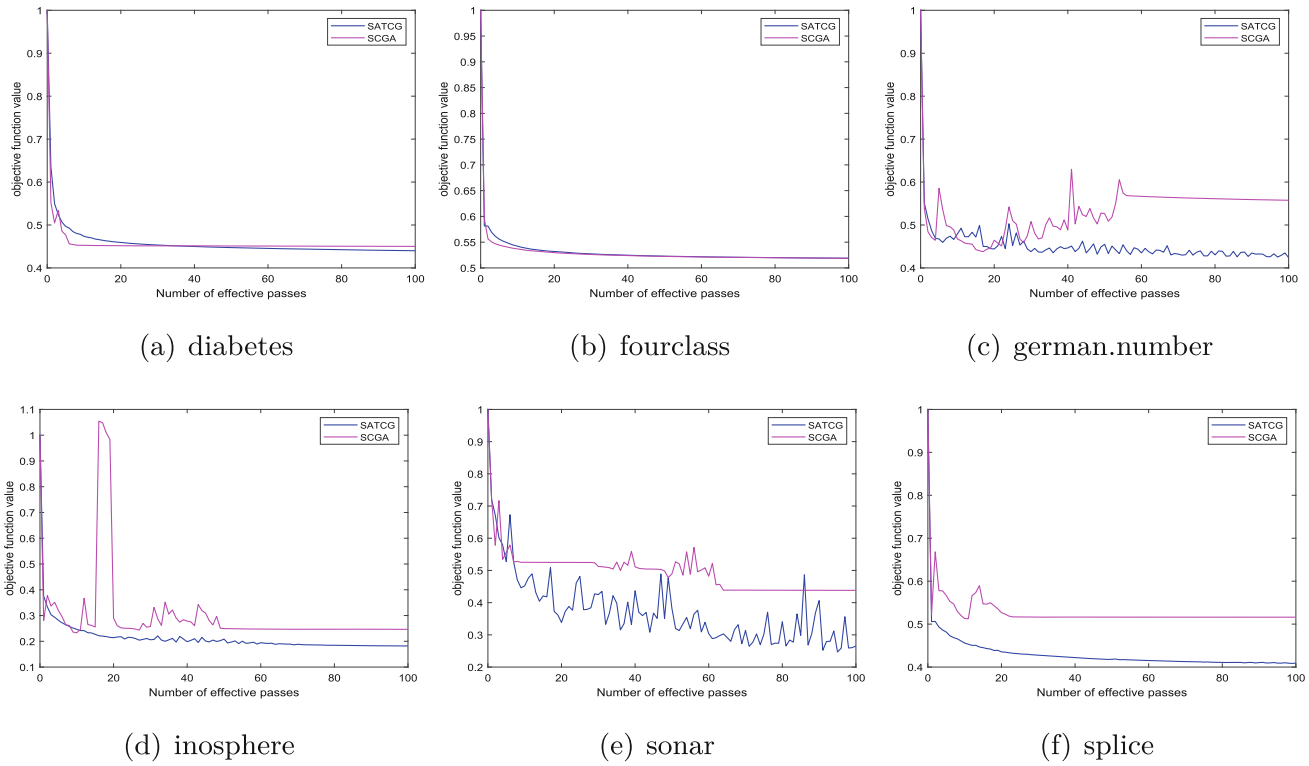


Fig. 3 Comparison of SATCG and SCGA on 6 datasets diabetes, fourclass, german.number, inosphere, snoar, splice for Model 1. (The x-axis represents numbers of iterations. The y-axis represents loss function values.)

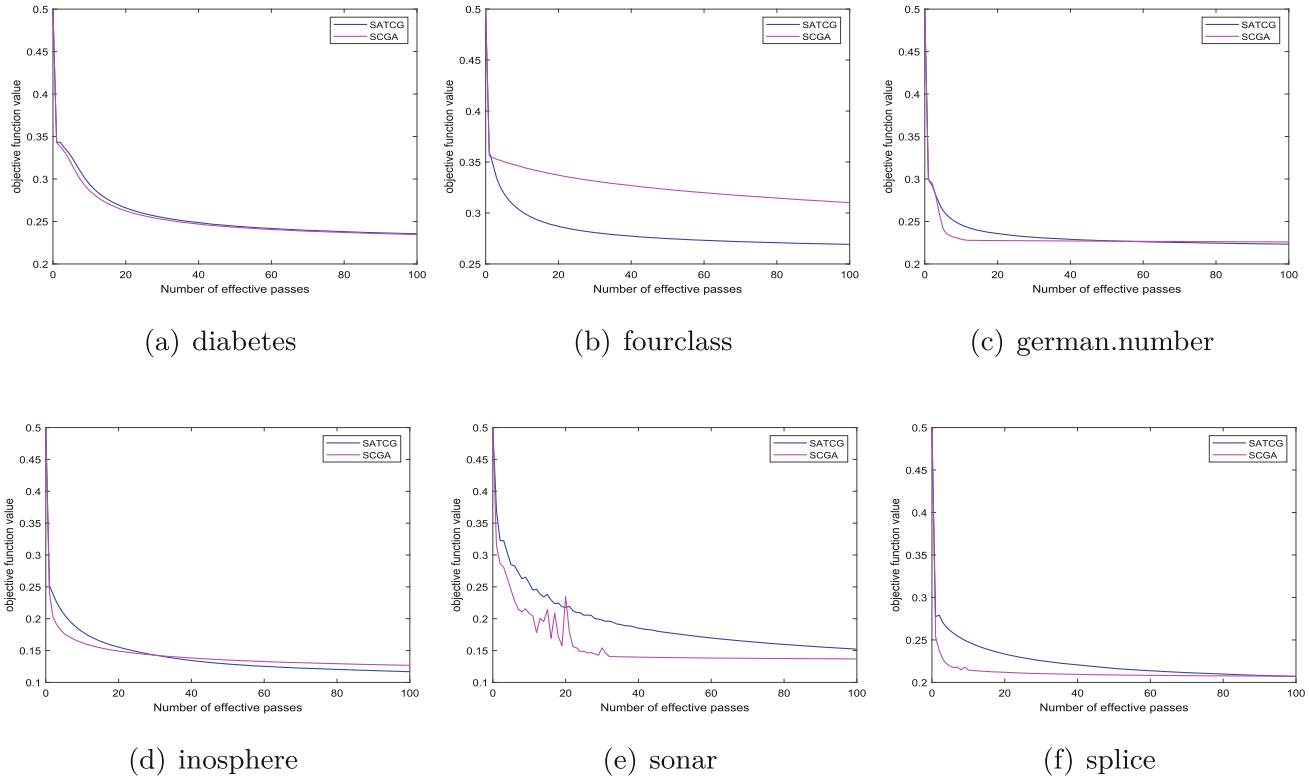


Fig. 4 Comparison of SATCG and SCGA on 6 datasets diabetes, fourclass, german.number, inosphere, snoar, splice for Model 2. (The x-axis represents numbers of iterations. The y-axis represents loss function values.)

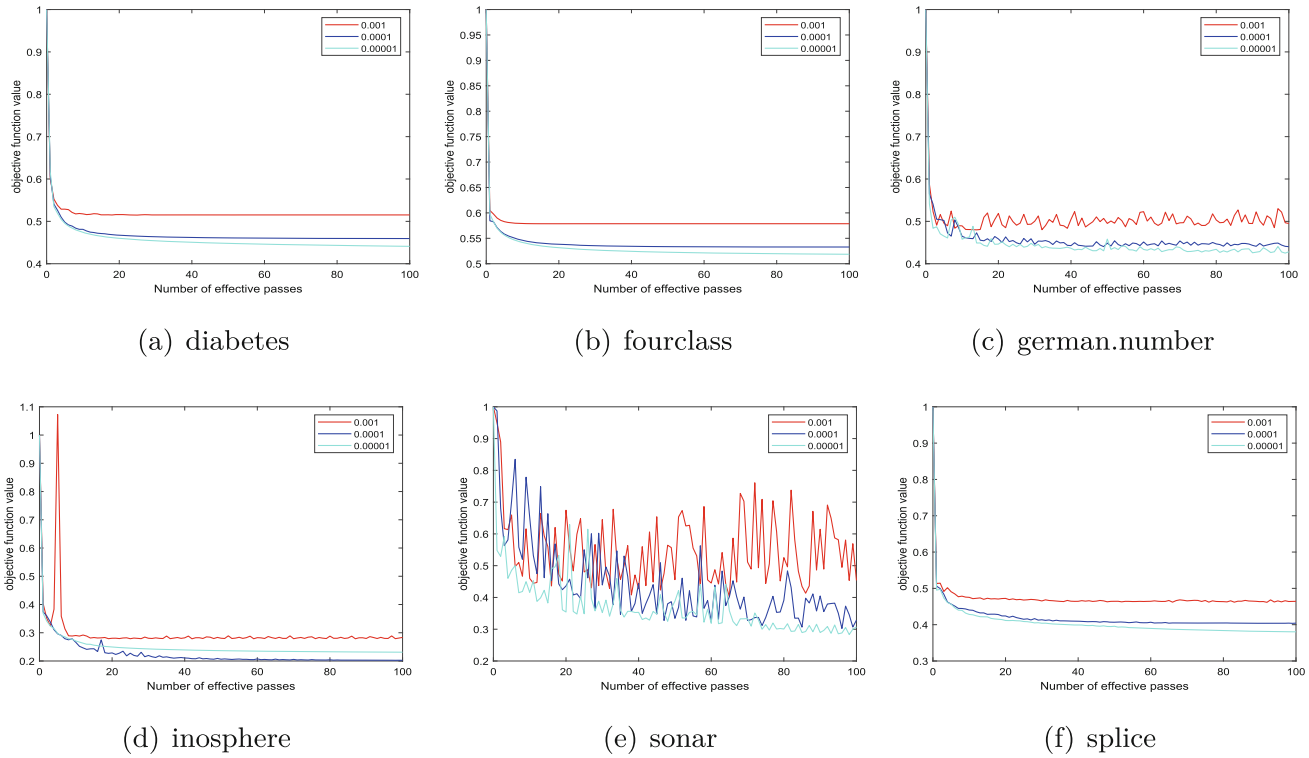


Fig. 5 Compare the performance of SATCG with different regularization coefficients (0.001,0.0001,0.00001) on Model 1. (The x-axis represents numbers of iterations. The y-axis represents loss function values.)

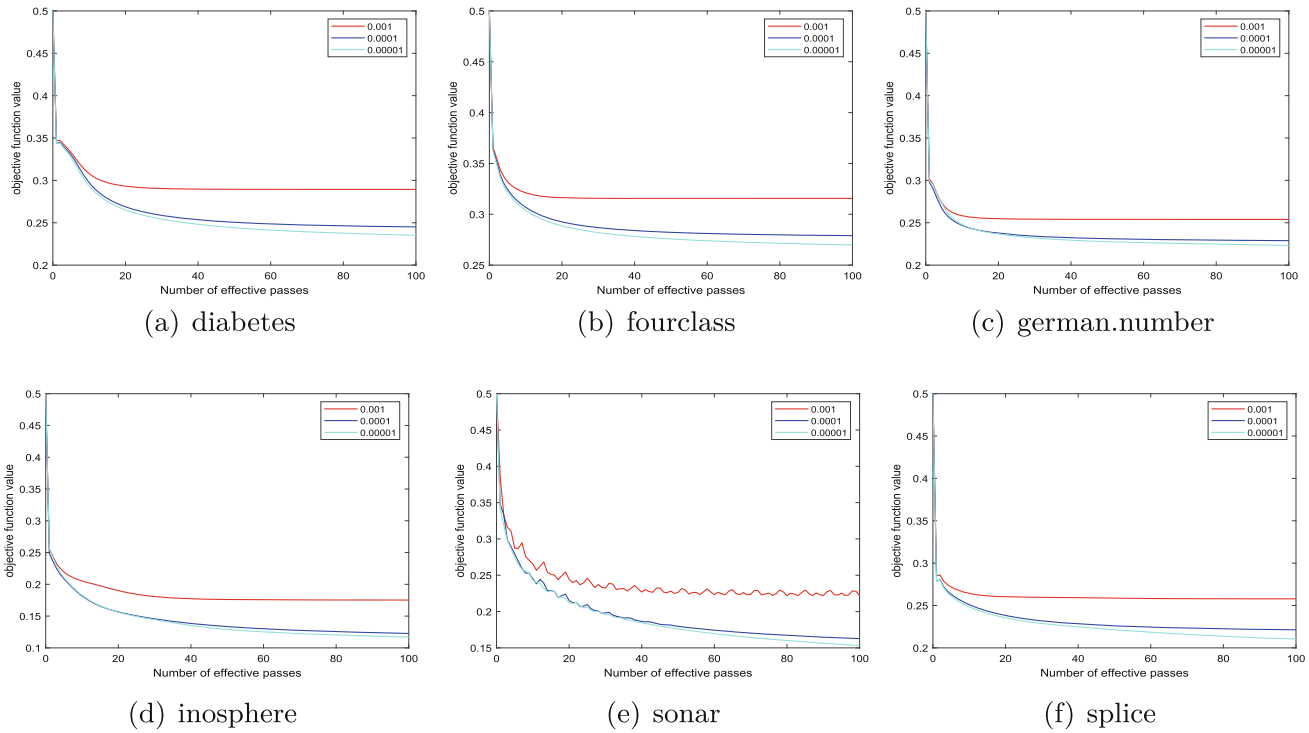


Fig. 6 Compare the performance of SATCG with different regularization coefficients (0.001,0.0001,0.00001) on Model 2. (The x-axis represents numbers of iterations. The y-axis represents loss function values.)

In summary, SATCG offers more advantages compared to SGD algorithms, and it exhibits greater stability and competitiveness than SCGA, Adam, and Rmsprop.

5 Conclusion

In this paper, we present an extension of the modified traditional three-conjugate gradient method and line search technique for immediate optimization. We achieve a linear convergence rate with weaker conditional assumptions in our theoretical analysis. Additionally, we test the convergence of our algorithm across 16 datasets using machine learning models. We compare the results of our algorithm SATCG with several other mainstream algorithms and find that SATCG demonstrates faster and more stable convergence.

Acknowledgements Thanks to the editor and reviewers for their constructive comments on the paper, which are of great help to improve the paper.

Author Contributions The contributions can be divided into the following parts: Gonglin Yuan: conceptualization, methodology, software, supervision. Chen Ouyang: data curation, writing-original draft preparation, writing-reviewing and editing, software, validation, visualization, investigation, formal analysis. Xiong Zhao: writing-original draft preparation, data curation. Rupin Huang: writing-review and editing. Yiyan Jiang: verify the attestation process. Chenkaixiang Lu corrected and supplemented the proofs and experiments of the article.

Funding This work is supported by Guangxi Science and Technology Base and Talent Project (Grant No. AD22080047) and the Innovation Funds of Chinese University (Grant No. 2021BCF03001).

Availability of data and materials All datasets can be found at https://datasetsearch.research.google.com and can be also found in the LIBSVM data website.

Declarations

Conflict of interest The authors declare no Conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Bottou, L.: Large-scale machine learning with stochastic gradient descent//Proceedings of COMPSTAT'2010: 19th international conference on computational statistics Paris France, Aug 22-27, 2010 Keynote, Invited and Contributed Papers. Physica-Verlag HD, (2010): 177-186

Bottou, L., Curtis, F.E., Nocedal, J.: Optimization methods for large-scale machine learning. SIAM Rev. **60**(2), 223–311 (2018)

Cauchy, A.: Méthode générale pour la résolution des systemes d'équations simultanées[J]. Comp. Rend. Sci. Paris **1847**(25), 536–538 (1847)

Dai, Y.H.: Conjugate gradient methods with Armijo-type line searches. Acta Math. Appl. Sin. **18**(1), 123–130 (2002)

Dai, Y.H., Yuan, Y.: A nonlinear conjugate gradient method with a strong global convergence property. SIAM J. Optim. **10**(1), 177–182 (1999)

Defazio, A., Bach, F., Lacoste-Julien S.: . SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives. Adv. Neural Inform. Process. Syst. (2014), 27

Fletcher, R., Reeves, C.M.: Function minimization by conjugate gradients. Comput. J. **7**(2), 149–154 (1964)

Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press, (2016)

Hager, W.W., Zhang, H.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. **16**(1), 170–192 (2005)

Huang, R., Qin, Y., Liu, K., Yuan, G. (2023). Biased stochastic conjugate gradient algorithm with adaptive step size for nonconvex problems. Expert Systems with Applications 121556

Jiang, X.Z., Zhu, Y.H., Jian, J.B.: Two efficient nonlinear conjugate gradient methods with restart procedures and their applications in image restoration[J]. Nonlinear Dyn. **111**(6), 5469–5498 (2023)

Jin, X.B., Zhang, X.Y., Huang, K., et al.: Stochastic conjugate gradient algorithm with variance reduction. IEEE Trans. Neural Netw. Learn. Syst. **30**(5), 1360–1369 (2018)

Johnson, R., Zhang, T.: Accelerating stochastic gradient descent using predictive variance reduction. Adv. Neural Inform. Process. Syst. (2013), 26

Karimi, H., Nutini, J., Schmidt, M. (2016). Linear convergence of gradient and proximal-gradient methods under the polyak–ojasiewicz condition. In Machine learning and knowledge discovery in databases: European conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I 16 (pp. 795-811). Springer International Publishing

Kim, H., Wang, C., Byun, H., et al.: Variable three-term conjugate gradient method for training artificial neural networks. Neural Netw. **159**, 125–136 (2023)

Kim, H., Wang, C., Byun, H., et al.: Variable three-term conjugate gradient method for training artificial neural networks. Neural Netw. **159**, 125–136 (2023)

Kingma D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, (2014)

Kou, C., Yang, H.: A mini-batch stochastic conjugate gradient algorithm with variance reduction. J. Glob. Optim. **87**, 1–17 (2022)

Lydia, A., Francis, S.: Adagrad-an optimizer for stochastic gradient descent. Int. J. Inf. Comput. Sci. **6**(5), 566–568 (2019)

Nguyen L.M., Liu, J., Scheinberg, K, et al. SARAH: a novel method for machine learning problems using stochastic recursive gradient//International conference on machine learning. PMLR, 2017: 2613-2621

Ouyang, L., Wu, J., Jiang, X., et al.: Training language models to follow instructions with human feedback. Adv. Neural. Inf. Process. Syst. **35**, 27730–27744 (2022)

Polak, E., Ribiere, G.: Note sur la convergence de méthodes de directions conjuguées[J]. Revue française d'informatique et de recherche opérationnelle. Série rouge, 1969, 3(16): 35-43

Polyak, B.T.: The conjugate gradient method in extremal problems. USSR Comput. Math. Math. Phys. **9**(4), 94–112 (1969)

Reddi, S.J, Kale, S., Kumar, S.: On the convergence of adam and beyond. arXiv preprint arXiv:1904.09237, (2019)

Robbins, H., Monro, S.: A stochastic approximation method. Ann. Math. Stat. 1951: 400-407

Schmidt, M., Le Roux, N., Bach, F.: Minimizing finite sums with the stochastic average gradient. Math. Program. **162**, 83–112 (2017)

Wang, X., Yuan, G., Pang, L.: A class of new three-term descent conjugate gradient algorithms for large-scale unconstrained opti-

mization and applications to image restoration problems. Numer Algorithms **93**(3), 949–970 (2023)
Wang B, Ye Q. Improving deep neural networks' training for image classification with nonlinear conjugate gradient-style adaptive momentum. IEEE Trans. Neural Netw. Learn. Syst. 2023
Wolfe, P.: Convergence conditions for ascent methods. SIAM Rev. **11**(2), 226–235 (1969)
Yang, Z.: Adaptive stochastic conjugate gradient for machine learning. Expert Syst. Appl. **206**, 117719 (2022)
Yuan, G., Wei, Z., Yang, Y.: The global convergence of the Polak-Ribière-Polyak conjugate gradient algorithm under inexact line search for nonconvex functions[J]. J. Comput. Appl. Math. **362**, 262–275 (2019)
Yuan, G., Lu, J., Wang, Z.: The modified PRP conjugate gradient algorithm under a non-descent line search and its application in the Muskingum model and image restoration problems. Soft. Comput. **25**(8), 5867–5879 (2021)
Yuan, G., Zhou, Y., Wang, L., et al.: Stochastic bigger subspace algorithms for nonconvex stochastic optimization. IEEE Access **9**, 119818–119829 (2021)
Yuan, G., Yang, H., Zhang, M.: Adaptive three-term PRP algorithms without gradient Lipschitz continuity condition for nonconvex functions. Num. Algorithms **91**(1), 145–160 (2022)
Yuan, G., Jian, A., Zhang, M., et al.: A modified HZ conjugate gradient algorithm without gradient Lipschitz continuous condition for non convex functions. J. Appl. Math. Comput. **68**(6), 4691–4712 (2022)
Zeiler M D. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701, 2012
Zhang, L., Zhou, W., Li, D.: Some descent three-term conjugate gradient methods and their global convergence. Optim. Methods Softw. **22**(4), 697–711 (2007)
Zhuang, J., Tang, T., Ding, Y., et al.: Adabelief optimizer: adapting stepsizes by the belief in observed gradients. Adv. Neural. Inf. Process. Syst. **33**, 18795–18806 (2020)