Advances in Applied Mathematics 应用数学进展, **2022, 11(7), 4248-4267**
Published Online July 2022 in Hans. http://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2022.117452

Hans汉斯

求解无约束优化问题的随机三项共轭梯度法

刘 蕾，薛 丹*

青岛大学，数学与统计学院，山东 青岛

收稿日期：2022年6月4日；录用日期：2022年6月29日；发布日期：2022年7月6日

摘 要

为了求解无约束随机优化问题，我们提出了一种带方差缩减的随机三项共轭梯度法(**STCGVR**)，此方法可以用来解决非凸随机问题。在算法的每次内循环迭代开始时，三项共轭梯度方向以最速下降方向重新开始迭代，有效地提高了收敛速度。在适当的条件下，讨论了该算法的性质和收敛性。数值结果表明，我们的方法对于求解机器学习问题具有巨大的潜力。

关键词

随机近似，经验风险最小化，三项共轭梯度，机器学习，方差缩减

A Stochastic Three-Term Conjugate Gradient Method for Unconstrained Optimization Problems

Lei Liu, Dan Xue*

School of Mathematics and Statistics, Qingdao University, Qingdao Shandong

Received: Jun. 4^{th}, 2022; accepted: Jun. 29^{th}, 2022; published: Jul. 6^{th}, 2022

* 通讯作者。

文章引用: 刘蕾, 薛丹. 求解无约束优化问题的随机三项共轭梯度法[J]. 应用数学进展, 2022, 11(7): 4248-4267.
DOI: 10.12677/aam.2022.117452

Abstract

To solve unconstrained stochastic optimization problems, a stochastic three-term conjugate gradient method with variance reduction (STCGVR) is proposed, which can be used to solve nonconvex stochastic problems. At the beginning of each inner loop iteration, the three conjugate gradient directions restart the iteration in the steepest descent direction, which effectively improves the convergence speed. The properties and convergence of the algorithm are discussed under appropriate conditions. The numerical results demonstrate that our method has dramatical potential for machine learning problems.

Keywords

Stochastic Approximation, Empirical Risk Minimization, Three-Term Conjugate Gradient, Machine Learning, Variance Reduction

1. 引言

我们考虑以下随机优化问题

$$\min_{x\in R^d} f(x) = E[F(x,\xi)], \tag{1}$$

这里$F: R^n \times R^d \to R$ 是连续可微的，并且可能非凸。ξ是一个随机变量。$E[\cdot]$ 表示对ξ的期望，$f(x) = E[F(x,\xi)]$ 被称为平均函数。由于在许多实际情况下，分布函数P未知或函数$F(\cdot,\xi)$未明确给出，其概率分布P 在支撑集$\Theta \subseteq R^d$上。为了得到目标函数值的一个比较好的估计，在实际问题中往往利用ξ的经验分布来代替实际分布。我们生成随机样本$\xi_1, \xi_2, ..., \xi_n$，令$f_i(x) = F(x,\xi_i)(i = 1, ..., n)$，得到经常出现在机器学习中的经验风险极小化问题

$$\min_{x\in R^d} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{2}$$

其中$f_i(x)$表示与第i个数据样本对应的损失函数，n表示数据样本数。问题(2)经常出现在机器学习 [1–6]以及无线系统中的最佳资源分配 [7,8] 中。

求解问题(2)时，存在一个挑战，即n可能非常大。由于精确的全梯度信息不容易获取，因此基于精确梯度的方法是不切实际的。为了克服这一困难，我们利用基于小样本数据的梯度近似方法，提出了随机梯度下降(SGD) [9]方法，此方法被视为求解大规模无约束优化问题的主要方法。在高维问题中，近似最优参数所需的迭代次数可能非常庞大，故SGD 方法的实际吸引力仍然有限。因此，一大批SGD的加速方法被提出。例如，随机平均梯度(SAG) [10,11]和SAGA [12]方法通过加入之前梯度值的记忆来实现更快的收敛速度，这些方法通常优于现有的SGD 方法。随机方差缩减梯度(SVRG) 方法 [13–15]有两个循环，在外循环中计算全梯度(每个外迭代称为一个历元)，在内循环中计算方差较小的随机梯度。S2GD [16,17]根据几何定律，在每个历元中运行随机数个随机梯度。此外，一些一阶方法，如AdaGrad [18]、RMSprop [19]和Adam [20] 也被证明在随机环境中是有效的。

为了解决目标函数曲率的问题，许多二阶的随机算法被提出，特别是BFGS算法。对于强凸问题，Mokhtari 和Ribeiro [21]提出了一种正则化随机BFGS(RES)方法，并给出了其收敛性分析。在 [22]中，Byrd 等人提出了一种基于随机逼近的随机有限记忆BFGS (L-BFGS) [23]方法，并证明了其对强凸问题的收敛性。Moritz 等人 [24]引入了L-BFGS 的一种随机变量，它结合了方差缩减的思想，因此对于强凸问题，它具有线性收敛速度。在 [25]中，Gower、Goldfarb和Richtarik提出了一种对凸函数线性收敛的方差缩减块L-BFGS方法。然而，在有限记忆随机拟牛顿方法中，经常需要m 个向量对来有效地计算乘积$H\nabla f$ (H是Hessian)。在内存有限的情况下，对于大规模机器学习问题可能是非常困难的。

共轭梯度(CG)方法结构简单，内存要求低，因此被广泛用于解决大规模优化问题 [26–28]。Fletcher和Reeves(FR) [26]首先提出了如何将线性共轭梯度法扩展到非线性函数，称为FR方法。在 [29] 中，Dai 和Liao提出了Dai-Liao三项共轭梯度法，并将拟牛顿技术与共轭性质相结合，获得了更好的收敛结果。此外，基于拟牛顿条件，Babaie, Kafaki 和Ghanbari [30]，Andrei [31]获得了一系列三项共轭梯度方法，这些方法对于强凸函数是全局收敛的。在文献 [32]中，Yao 提出了一种改进的Dai-Liao三项共轭梯度法。本文在文献 [32]的基础上，提出了一种带方差减小的随机三项共轭梯度法(STCGVR)，它将改进的Dai-Liao 三项共轭梯度与随机方差缩减相结合，用于求解无约束随机优化问题。

我们在本文中的贡献如下：

1. 基于SVRG的最新进展，提出了求解随机优化问题(2)的STCGVR方法，并证明了其对强凸光滑函数的线性收敛性。

2. 在STCGVR的每次内循环开始时，重新启动最速下降的迭代方向，有效地提高了收敛速度。

3. 对几个机器学习问题的数值实验表明，与SVRG方法相比，STCGVR方法是非常有效的。

本文的剩余部分组织如下。第2节介绍了用于解决随机优化问题的改进的Dai-Liao三项共轭梯度法、SVRG算法和带方差缩减的随机三项共轭梯度法。在第3节中，在适当的条件下证明了新算法的收敛性。在第4节中，报告了一些初步的数值结果。最后，第5节得出一些结论。

2. 用于无约束优化的STCGVR的算法

2.1. 三项共轭梯度

共轭梯度法因其简单且存储量低而被广泛用于解决大规模优化问题，它会生成一系列迭代：

$$x_{k+1} = x_k + \alpha_k d_k, \tag{3}$$

这里步长α_k 由以下Wolfe 线搜索确定:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k g_k^T d_k, \tag{4}$$

$$g_{k+1}^T d_k \geq c_2 g_k^T d_k, \tag{5}$$

这里$0 < c_1 < c_2 < 1$, 搜索方向d_k 由以下公式确定：

$$d_k = \begin{cases} -g_0, & k = 0. \\ -g_k + \beta_k d_{k-1}, & k \geq 1. \end{cases} \tag{6}$$

其中，β_k是一个参数，而$g_k = \nabla f(x_k)$是目标函数$f(x)$在x_k 处的梯度。共轭梯度法最典型的特征是共轭性，即(6)生成的搜索方向应具有以下共轭条件：

$$d_{k+1} y_k = 0, k \geq 1, \tag{7}$$

其中，$y_k = g_{k+1} - g_k$。近年来，共轭条件一直是研究者关注的焦点。Dai 和Liao 获得了一个显著的结果 [29]。在 [29]中，使用标准割线方程：

$$B_{k+1} s_k = y_k, \tag{8}$$

其中$s_k = x_{k+1} - x_k$，B_{k+1}是$f(x)$Hessian阵的近似对称矩阵。然后将共轭条件(7) 推广到Dai-Liao共轭条件

$$d_{k+1}^T y_k = -t_1 g_{k+1}^T s_k, \tag{9}$$

其中t_1为非负参数。基于Dai-Liao共轭条件(9)和拟牛顿技术，Yao等人提出了一种对称修正的Dai-Liao矩阵

$$Q_{t+1}^{MP} = I + \eta_k Q_2^{k+1} + Q_1^{k+1}, \tag{10}$$

其中

$$Q_2^{k+1} = -\frac{s_k y_k^T - y_k s_k^T}{s_k^T y_k}, Q_1^{k+1} = \frac{s_k s_k^T}{s_k^T y_k}, \tag{11}$$

其中η_k是待确定的正参数。搜索方向由以下公式生成

$$d_{k+1} = -Q_{k+1}^{MP} g_{k+1}, k \geq 1, \tag{12}$$

根据Q_{k+1}^{MP}的定义，由(12)生成的搜索方向取决于每次迭代时的参数η_k。由于该方法的原理是由(12)生成的搜索方向应满足Dai-Liao 共轭条件(9)。根据该原理，结合(10)、(11)和(12)，我们得到

$$\eta_k = \frac{g_{k+1}^T y_k + (1-t_1) g_{k+1}^T s_k}{g_{k+1}^T y_k - \frac{||y_k||^2}{s_k^T y_k} g_{k+1}^T s_k}, \tag{13}$$

其中，参数t_1是(9)中的Dai-Liao参数t_1。

另一方面，从η_k的定义来看，η_k的值可能非常大，甚至趋于无穷大。为了获得算法的全局收敛性，η_k被限制如下：

$$\eta_k = min\{|\frac{g_{k+1}^T y_k + (1-t_1) g_{k+1}^T s_k}{g_{k+1}^T y_k - \frac{||y_k||^2}{s_k^T y_k} g_{k+1}^T s_k}|, M_1\}, \tag{14}$$

其中M_1为正常数。实际上，由$(10)-(12)$生成的方向可以重写为典型的三项共轭梯度方向：

$$d_{k+1} = -g_{k+1} + \beta_k d_k + \delta_k y_k, \tag{15}$$

其中t_1, β_k, δ_k 由

$$t_1 = \frac{||y_k||^2}{s_k^T y_k}, \tag{16}$$

$$\beta_k = max\{\frac{\eta_k g_{k+1}^T y_k - g_{k+1}^T s_k}{d_k^T y_k}, 0\}, \tag{17}$$

$$\delta_k = -\eta_k \frac{g_{k+1}^T s_k}{y_k^T s_k}, \tag{18}$$

生成，其中参数η_k由(14)确定。

改进的三项共轭梯度法将共轭条件与拟牛顿技术相结合，有效地提高了传统共轭梯度法的效率。因此，该方法在求解大规模优化问题中具有很大的发展前景。

2.2. 随机方差缩减梯度(SVRG)算法

在SGD中，单个样本的梯度是全体样本平均梯度的一个无偏估计，但梯度的方差会随着迭代

的增加不断累加，这会使得SGD的收敛速度变慢，无法达到线性收敛，所以我们引入了方差缩减策略。方差缩减策略通过构造特殊的随机梯度估计量，使得每次迭代的方差有一个不断缩减的上界，从而取得较快的收敛速度。

我们提出了优化(2)的SVRG算法，并在算法1中对其进行了描述。

Algorithm 1. SVRG

初始化:
给定一个初始点$\tilde{x}_0 \in R^n$, 固定步长α，选取常数$0 < c_1 < c_2 < 1$, $M_1 > 0$.

for k=0,1,2,... **do**
  $x_0^{k+1} = \tilde{x}_k$.
  计算全梯度$\nabla f(\tilde{x}_k) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\tilde{x}_k)$.
  for t=0,1,...,m-1 **do**
    从$i_t \subset \{1, 2, ..., n\}$ 中随机抽取一个样本.
    计算随机梯度
      $g_t^{k+1} = \nabla f_{i_t}(x_t^{k+1}) - (\nabla f_{i_t}(\tilde{x}_k) - \nabla f(\tilde{x}_k))$.
    计算$x_{t+1}^{k+1} = x_t^{k+1} - \alpha g_t^{k+1}$.
  end for
  $\tilde{x}_{k+1} = \frac{1}{m}\sum_{t=1}^{m} x_t^{k+1}$.
end for

算法1中有两个循环。在外循环中，全梯度$\nabla f(\tilde{x}_k)$被计算。$\tilde{x}_k$每隔m 次更新保存一个“快照”，记为x_0^k。在内循环中，我们从数据集X 中随机选择一个样本用于生成随机梯度。也就是说，设$\tilde{x}_k$为第k 个检查点，然后我们需要计算$\tilde{x}_k$ 点处的全梯度：

$$\nabla f(\tilde{x}_k) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\tilde{x}_k). \tag{19}$$

在后续迭代中，方向g_t^{k+1}用作更新方向：

$$g_t^{k+1} = \nabla f_{i_t}(x_t^{k+1}) - (\nabla f_{i_t}(\tilde{x}_k) - \nabla f(\tilde{x}_k)), \tag{20}$$

其中$i_t \subset \{1, 2, ..., N\}$ 被任意抽取。注意到随机梯度g_{t+1}^{k+1} 是$\nabla f(x_{t+1}^{k+1})$的一个无偏梯度估计，即：$E[g_{t+1}^{k+1} | x_{t+1}^{k+1}] = \nabla f(x_{t+1}^{k+1})$.

2.3. STCGVR 算法

本文的目标是设计一种使梯度方差较低的方法，同时具有低内存要求。为此，我们将SVRG与改进的Dai-Liao三项共轭梯度法相结合。算法2中总结了STCGVR算法。在算法2中，我们通过以下迭代计算搜索方向d_{t+1}:

$$d_{t+1} = -g_{t+1}^{k+1} + \beta_t d_t + \delta_t y_t, \tag{21}$$

where t_1, β_t, δ_t by

$$t_1 = \frac{||y_t||^2}{s_t^T y_t}, \tag{22}$$

$$\beta_t = max\{\frac{\eta_t (g_{t+1}^{k+1})^T y_t - (g_{t+1}^{k+1})^T s_t}{d_t^T y_t}, 0\}, \tag{23}$$

$$\delta_k = -\eta_t \frac{(g_{t+1}^{k+1})^T s_t}{y_t^T s_t}, \tag{24}$$

$$\eta_t = min\{|\frac{(g_{t+1}^{k+1})^T y_t + (1-t_1)(g_{t+1}^{k+1})^T s_t}{(g_{t+1}^{k+1})^T y_t - \frac{||y_t||^2}{s_t^T y_t}(g_{t+1}^{k+1})^T s_t}|, M_1\}, \tag{25}$$

其中M_1 是一个正常数，$s_t = x_{t+1}^{k+1} - x_t^{k+1}, y_t = g_{t+1}^{k+1} - g_t^{k+1}$.

Algorithm 2. STCGVR

初始化:
给定一个初始点$\tilde{x}_0$, 初始步长α_0, 更新频率m, 迭代$\{x_t^{k+1} : t = 0, ..., m-1; k = 0, 1, 2, ...\}$, 选取常数$0 < c_1 < c_2 < 1$, $M_1 > 0$.

$h_0 = \nabla f(\tilde{x}_0)$.
for k=0,1,2,... **do**
  计算全梯度$\nabla f(\tilde{x}_k) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\tilde{x}_k)$.
  令$x_0^{k+1} = \tilde{x}_k$, $g_0^{k+1} = h_k$, $d_0 = -g_0^{k+1}$.
  for t=0,1,...,m-1 **do**
    调用线搜索算法(4) and (5) 计算α_t.
    计算$x_{t+1}^{k+1} = x_t^{k+1} + \alpha_t d_t$.
    随机抽取$i_t \subset \{1, 2, ..., n\}$.
    计算随机梯度
      $g_t^{k+1} = \nabla f_{i_t}(x_t^{k+1}) - (\nabla f_{i_t}(\tilde{x}_k) - \nabla f(\tilde{x}_k))$.
    通过(21) − (25)计算d_{t+1}.
  end for
  $h_{k+1} = g_m^{k+1}$, $\tilde{x}_{k+1} = \frac{1}{m}\sum_{t=1}^{m} x_t^{k+1}$.
end for

与SVRG 类似，算法2分为两个循环。在外部循环中，计算外部迭代$\tilde{x}_k \in R^n$ 和完整梯度$\nabla f(\tilde{x}_k)$。在内循环中，使用SVRG 更新对梯度g_t^{k+1} 估计。此外，我们在每个内部循环迭代开始时以最速下降步重新开始迭代，例如： [33, 34]。重新启动将定期重启算法并消除可能无益的旧信息。因此，STCGVR可用于解决大规模无约束随机优化问题，具有良好的发展前景。

3. 收敛性分析

在本节中，我们证明算法2生成的迭代序列是线性收敛的。

下面我们给出本文中要用到的一些假设。

假设1 假设水平集$F = \{x|f(x) \leq f(x_0)\}$是有界的。此外，函数$f(x)$ 在F中是有界的。

假设2 假设$f_i : R^n \to R$ 连续可微，∇f 是全局Lipschitz 连续的，其Lipschitz 常数为L，即：对于$\forall x, y \in R^n$,有以下成立：

$$||\nabla f(x) - \nabla f(y)|| \leq L||x - y||. \tag{26}$$

假设3 STCGVR算法中的步长α_t满足$\alpha_t \in [\alpha_l, \alpha_r]$ $(0 < \alpha_l < \alpha_r)$.

假设4 由于随机梯度g_t^{k+1}是$\nabla f(x_t^{k+1})$的一个无偏估计，即：$E[g_t^{k+1}|x_t^{k+1}] = \nabla f(x_t^{k+1})$, 故存在一个正常数H, 对于所有$t = 0, 1, ..., m-1$; $k = 0, 1, 2, ...$，有

$$||\nabla f(x_t^{k+1}) - g_t^{k+1}|| \leq H. \tag{27}$$

假设5 存在两个正常数$\underline{\kappa}, \overline{\kappa}$ ，有以下成立：

$$\underline{\kappa} I \preceq Q_t^{MP} \preceq \overline{\kappa} I, \forall t, \tag{28}$$

其中符号$A \succeq B$, $A, B \in R^{n\times n}$代表$A - B$是半正定的。

假设6 对于所有$t = 0, 1, ..., m-1$; $k = 0, 1, 2, ...$，随机梯度g_t^{k+1} 是有界的，即：

$$||g_t^{k+1}|| \leq \Lambda. \tag{29}$$

引理3.1. *假设d_t由$(21) - (25)$生成，如果步长α_t是由Wolfe 搜索条件(4) 和(5)生成，则充分下降性质对任何$t = 0, ..., m-1$; $k = 0, 1, 2, ...$成立；即存在一个正常数ρ_1，使得*

$$-(g_t^{k+1})^T d_t \geq \rho_1 ||g_t^{k+1}||^2. \tag{30}$$

证明：因为$(g_0^{k+1})^T d_0 = -||g_0^{k+1}||^2$, 充分下降条件对于$t = 0$成立. 由等式$(21) - (25)$, 我们得到

$$\begin{aligned}
(d_{t+1}^{k+1})^T g_{t+1}^{k+1} &= (-g_{t+1}^{k+1} + \beta_t d_t + \delta_t y_t)^T g_{t+1}^{k+1} \\
&= -||g_{t+1}^{k+1}||^2 + \beta_t (g_{t+1}^{k+1})^T d_t + \delta_t (g_{t+1}^{k+1})^T y_t \\
&= -||g_{t+1}^{k+1}||^2 + \frac{\eta_t (g_{t+1}^{k+1})^T y_t - (g_{t+1}^{k+1})^T s_t}{s_t^T y_t} (g_{t+1}^{k+1})^T s_t - \eta_t \frac{(g_{t+1}^{k+1})^T s_t}{y_t^T s_t} (g_{t+1}^{k+1})^T y_t \\
&= -||g_{t+1}^{k+1}||^2 - \frac{((g_{t+1}^{k+1})^T s_t)^2}{s_t^T y_t}
\end{aligned} \tag{31}$$

此外, Wolfe线搜索条件(4) 和(5) 可以确保$s_t^T y_t > 0$. 故等式(31) 意味着充分下降条件对于$\rho_1 = 1$成立。

搜索方向的充分下降特性在STCGVR 的收敛性分析中是必不可少的。上述引理3.1 表明算法2生成的搜索方向在Wolfe 线搜索下具有该性质。

引理3.2. 假设d_t由*(21)-(25)*生成，如果步长α_t具有*Wolfe*线搜索条件(4) 和(5)确定，$f(x)$满足假设*2*和假设*4*, 我们得：

$$\alpha_t \geq \frac{(c_2-1)(g_t^{k+1})^T d_t - 2H||d_t||}{L||d_t||^2}. \tag{32}$$

证明：由假设2和假设4, 我们得到

$$\begin{aligned} ||y_t|| &= ||g_{t+1}^{k+1} - g_t^{k+1}|| \\ &= ||g_{t+1}^{k+1} - \nabla f(x_{t+1}^{k+1}) + \nabla f(x_{t+1}^{k+1}) - \nabla f(x_t^{k+1}) + \nabla f(x_t^{k+1}) - g_t^{k+1}|| \\ &\leq ||g_{t+1}^{k+1} - \nabla f(x_{t+1}^{k+1})|| + ||\nabla f(x_{t+1}^{k+1}) - \nabla f(x_t^{k+1})|| + ||\nabla f(x_t^{k+1}) - g_t^{k+1}|| \\ &\leq 2H + L||s_t||, \end{aligned} \tag{33}$$

结合Lipschitz不等式(27)和Wolfe条件，我们能推出

$$\begin{aligned} (c_2-1)(g_t^{k+1})^T d_t &\leq (g_{t+1}^{k+1} - g_t^{k+1})^T d_t \\ &= y_t^T d_t \\ &\leq ||y_t||||d_t|| \\ &\leq (2H + L||s_t||)||d_t||. \end{aligned} \tag{34}$$

因此，引理得证。

引理3.3. 假设d_t由$(21)-(25)$生成，如果步长α_t具有*Wolfe*线搜索条件(4) 和(5)确定，$f(x)$满足假设*1*和假设*2*, 那么以下*Zoutendijk* 条件成立：

$$\sum_{t\geq 1} \frac{((g_t^{k+1})^T d_t)^2}{||d_t||^2} < \infty. \tag{35}$$

证明：由Wolfe条件(4)得

$$f(x_t^{k+1}) - f(x_{t+1}^{k+1}) \geq -c_1\alpha_t(g_t^{k+1})^T d_t, \tag{36}$$

结合(32)，我们有

$$\begin{aligned} f(x_t^{k+1}) - f(x_{t+1}^{k+1}) &\geq -c_1 \frac{(c_2-1)(g_t^{k+1})^T d_t - 2H||d_t||}{L||d_t||^2}(g_t^{k+1})^T d_t \\ &\geq \frac{c_1(1-c_2)((g_t^{k+1})^T d_t)^2 + 2c_1 H||d_t||(g_t^{k+1})^T d_t}{L||d_t||^2}, \end{aligned} \tag{37}$$

通过对(37)两边取绝对值并求和，得到

$$\begin{aligned}&\sum_{t\geq 1}|f(x_t^{k+1})-f(x_t^{k+1}+\alpha_t d_t)|\\&\geq\sum_{t\geq 1}|\frac{c_1(1-c_2)((g_t^{k+1})^T d_t)^2}{L||d_t||^2}+\frac{2c_1H||d_t||(g_t^{k+1})^T d_t}{L||d_t||^2}|\\&\geq\sum_{t\geq 1}(|\frac{c_1(1-c_2)((g_t^{k+1})^T d_t)^2}{L||d_t||^2}|-|\frac{2c_1H||(g_t^{k+1})||}{L}|).\end{aligned}\tag{38}$$

通过对式(38)两边求和，并结合假设1和$\|g_t^{k+1}\|$有界，则得到Zoutendijk 条件在随机情况下成立：

$$\sum_{t\geq 1}\frac{((g_t^{k+1})^T d_t)^2}{||d_t||^2}<\infty.\tag{39}$$

引理3.4. 假设d_t由$(21)-(25)$生成，如果步长α_t具有*Wolfe*线搜索条件(4) 和(5)确定，那么对于强凸函数，序列d_t的范数是有界的，即存在$M>0$ 使得

$$||d_t||\leq M,\tag{40}$$

成立。

证明：由等式$(10)-(12)$,假设5和假设6，我们得

$$||d_t||=||-Q_t^{MP}g_t^{k+1}||\leq\overline{\kappa}||g_t^{k+1}||\leq\overline{\kappa}\Lambda=M.\tag{41}$$

定理3.1. 假设d_t由$(21)-(25)$生成，如果步长α_t具有*Wolfe*线搜索条件(4) 和(5)确定，目标函数$f(x)$是强凸的并且满足假设*1*，那么我们有

$$\lim_{t\to\infty}||g_t^{k+1}||=0.\tag{42}$$

证明：基于引理3.4，我们得到$\|d_t\leq M\|$。根据充分下降条件：$-(g_t^{k+1})^T d_t\geq\rho_1\|g_t^{k+1}\|^2$，再结合引理3.3，我们有

$$\infty>\sum_{t\geq 1}\frac{((g_t^{k+1})^T d_t)^2}{||d_t||^2}\geq\sum_{t\geq 1}\frac{((g_t^{k+1})^T d_t)^2}{M^2}\geq\frac{\rho_1^2}{M^2}\sum_{t\geq 1}||g_t^{k+1}||^2,\tag{43}$$

从而推出(42)，定理得证。

以上定理3.1表明我们提出的算法对于强凸函数是全局收敛的。

引理3.5. 假定假设*2*成立，x_*是$f(x)$的唯一最小值点。那么对于任意$x\in R^n$，我们有

$$\frac{1}{2L}||\nabla f(x)||^2\leq f(x)-f(x_*).\tag{44}$$

证明：由于x_*是$f(x)$的唯一极小值点，由假设2，我们有

$$f(x_*) \leq f(y) \leq f(x) + \nabla f(x)^T(y-x) + \frac{L}{2}||y-x||^2, \tag{45}$$

固定x, 因为以上公式对于任意的y都成立，故下确界可以在上述不等式的右边得到：

$$\begin{aligned} f(x_*) &\leq f(x) + \nabla f(x)^T(y-x) + \frac{L}{2}||y-x||^2 \\ &= f(x) - \frac{L}{2}||\nabla f(x)||^2. \end{aligned} \tag{46}$$

因此，不等式(44)成立。

引理3.6. 假定x_*为$f(x)$的唯一极小值点，假设2成立，$g_t^{k+1} = \nabla f_{i_t}(x_t^{k+1}) - (\nabla f_{i_t}(\tilde{x}_k) - \nabla f(\tilde{x}_k))$是方差减小的随机梯度。相对于$i_t$取期望，我们得到

$$E[||g_t^{k+1}||^2] \leq 4L(E[f(x_t^{k+1}) - f(x_*)] + E[f(\tilde{x}_k) - f(x_*)]). \tag{47}$$

证明：由g_t^{k+1}的更新公式，我们得到

$$\begin{aligned} E[||g_t^{k+1}||^2] &= E[||\nabla f_{i_t}(x_t^{k+1}) - \nabla f_{i_t}(\tilde{x}_k) + \nabla f(\tilde{x}_k)||^2] \\ &= E[||\nabla f_{i_t}(x_t^{k+1}) - \nabla f_{i_t}(\tilde{x}_k) + \nabla f(\tilde{x}_k) + \nabla f_{i_t}(x_*) - \nabla f_{i_t}(x_*)||^2] \\ &\leq 2E[||\nabla f_{i_t}(x_t^{k+1}) - \nabla f_{i_t}(x_*)||^2] + 2E[||\nabla f_{i_t}(\tilde{x}_k) - \nabla f(\tilde{x}_k) - \nabla f_{i_t}(x_*)||^2]. \end{aligned} \tag{48}$$

接下来，我们构造一个辅助函数：

$$\Phi_i(x) = f_i(x) - f_i(x_*) - \nabla f_i(x_*)(x - x_*), \tag{49}$$

注意到$\Phi_i(x)$是一个凸函数，∇f是全局Lipschitz连续的，其Lipschitz连续常数为L，结合(44)，我们有

$$\frac{1}{2L}||\nabla \Phi_i(x)||^2 \leq \Phi_i(x) - \Phi_i(x_*). \tag{50}$$

应用$\Phi_i(x)$和$\nabla \Phi_i(x)$的表达式，我们得到

$$||\nabla f_i(x) - f_i(x_*)||^2 \leq 2L[f_i(x) - f_i(x_*) - \nabla f_i(x_*)(x - x_*)]. \tag{51}$$

对(51)两边从1到n求和，并且注意到$\nabla f(x_*) = 0$，我们得到

$$\frac{1}{n}\sum_{i=1}^{n}||\nabla f_i(x) - f_i(x_*)||^2 \leq 2L[f(x) - f(x_*)], \forall x. \tag{52}$$

因此，我们有

$$\begin{aligned} &E[||\nabla f_{i_t}(x_t^{k+1}) - \nabla f_{i_t}(x_*)||^2] \\ &= E[E[||\nabla f_{i_t}(x_t^{k+1}) - \nabla f_{i_t}(x_*)||^2]] \\ &= E[\frac{1}{n}\sum_{i=1}^{n}||\nabla f_i(x_t^{k+1}) - \nabla f_i(x_*)||^2] \\ &\leq 2LE[f(x_t^{k+1}) - f(x_*)], \end{aligned} \tag{53}$$

和

$$E[||\nabla f_{i_t}(\tilde{x}_{k-1}) - \nabla f_{i_t}(x_*)||^2] \leq 2LE[f(\tilde{x}_{k-1}) - f(x_*)]. \tag{54}$$

再结合(47), (53), (54)，我们有

$$E[||g_t^{k+1}||^2] \leq 4L(E[f(x_t^{k+1}) - f(x_*)] + E[f(\tilde{x}_k) - f(x_*)]). \tag{55}$$

定理3.2. 假定假设*1*-假设*4* 成立并且$f(x)$ 是强凸的，其强凸参数是u, 令x^*是$f(x)$ 的唯一极小值，并假设m 足够大，使得

$$\rho = \frac{(\frac{2}{u} + 4L\alpha_r^2 m)}{2\alpha_l m(\underline{\kappa} + 2\alpha_l L\overline{\kappa}^2)} < 1. \tag{56}$$

那么，对于所有$k \geq 0$,我们有

$$E[f(\tilde{x}_k) - f(x^*)] \leq \rho^k E[f(\tilde{x}_0) - f(x^*)]. \tag{57}$$

证明：定义$\Delta_t = \|x_t^{k+1} - x^*\|$，又由等式(12)，假设2和假设5，我们有

$$\begin{aligned} E[\Delta_{t+1}^2] &= E[||x_{t+1}^{k+1} - x^*||^2] \\ &= E[||x_t^{k+1} - \alpha_t Q_t^{MP} g_t^{k+1} - x^*||^2] \\ &= E[\Delta_t^2] - 2\alpha_t E[< Q_t^{MP} g_t^{k+1}, x_t^{k+1} - x^* >] + \alpha_t^2 E[||Q_t^{MP} g_t^{k+1}||^2] \\ &= E[\Delta_t^2] - 2\alpha_t E[< Q_t^{MP}\nabla f(x_t^{k+1}), x_t^{k+1} - x^* >] + \alpha_t^2 ||Q_t^{MP}||^2 E[||g_t^{k+1}||^2] \\ &\leq E[\Delta_t^2] - 2\alpha_t \underline{\kappa}[f(x_t^{k+1}) - f(x^*)] + \alpha_t^2 \overline{\kappa}^2 E[||g_t^{k+1}||^2], \end{aligned} \tag{58}$$

结合(47)和引理3.6，我们得到

$$\begin{aligned} E[\Delta_{t+1}^2] &\leq E[\Delta_t^2] - 2\alpha_t \underline{\kappa}[f(x_t^{k+1}) - f(x^*)] \\ &\quad + 4\alpha_t^2 \overline{\kappa}^2 L(E[f(x_t^{k+1}) - f(x^*)] + E[f(\tilde{x}_k) - f(x^*)]) \\ &= E[\Delta_t^2] - (2\alpha_t \underline{\kappa} - 4\alpha_t^2 L\overline{\kappa}^2)[f(x_t^{k+1}) - f(x^*)] + 4\alpha_t^2 \overline{\kappa}^2 LE[f(\tilde{x}_k) - f(x^*)]. \end{aligned} \tag{59}$$

对t从0到m-1求和，结合$x_1^{k+1}=\tilde{x}_{k-1}$，我们有

$$\begin{aligned}&E[\Delta_{m+1}^2]+2\alpha_t(\underline{\kappa}+2\alpha_t L\overline{\kappa}^2)\sum_{t=1}^{m}E[f(x_t^{k+1})-f(x^*)]\\&\leq E[||\tilde{x}_{k-1}-x^*||^2]+4\alpha_t^2\overline{\kappa}^2LmE[f(\tilde{x}_{k-1})-f(x^*)].\end{aligned}\tag{60}$$

由于$f(x)$是强凸的，我们得到

$$\begin{aligned}&E[\Delta_{m+1}^2]+2\alpha_t(\underline{\kappa}+2\alpha_t L\overline{\kappa}^2)\sum_{t=1}^{m}E[f(x_t^{k+1})-f(x^*)]\\&\leq\frac{2}{u}E[f(\tilde{x}_{k-1})-f(x_*)]+4\alpha_t^2\overline{\kappa}^2LmE[f(\tilde{x}_{k-1})-f(x_*)].\end{aligned}\tag{61}$$

根据$\tilde{x}_{k+1}=\frac{1}{m}\sum_{t=1}^{m}x_t^{k+1}$，我们有

$$\begin{aligned}E[f(\tilde{x}_k)-f(x^*)]&\leq\frac{1}{m}\sum_{t=1}^{m}E[f(x_t^{k+1})-f(x^*)]\\&\leq\frac{1}{2\alpha_t(\underline{\kappa}+2\alpha_t L\overline{\kappa}^2)m}(\frac{2}{u}+4L\overline{\kappa}^2m\alpha_t^2)E[f(\tilde{x}_{k-1})-f(x^*)]\\&\leq\frac{1}{2\alpha_l(\underline{\kappa}+2\alpha_l L\overline{\kappa}^2)m}(\frac{2}{u}+4L\overline{\kappa}^2m\alpha_r^2)E[f(\tilde{x}_{k-1})-f(x^*)].\end{aligned}\tag{62}$$

通过归纳，我们有

$$E[f(\tilde{x}_k)-f(x^*)]\leq\rho^kE[f(\tilde{x}_0)-f(x^*)],\tag{63}$$

其中，

$$\rho=\frac{(\frac{2}{u}+4L\alpha_r^2m)}{2\alpha_lm(\underline{\kappa}+2\alpha_lL\overline{\kappa}^2)}.\tag{64}$$

如果令$\rho<1$，则有

$$m\geq\frac{1}{(\alpha_l\underline{\kappa}+2\alpha_l^2L\overline{\kappa}^2-2\alpha_r^2L\overline{\kappa}^2)u}.\tag{65}$$

定理3.2 意味着STCGVR 在函数值期望的意义上对于参考点$\tilde{x}_k$是线性收敛的。

在继续之前，我们介绍马尔可夫不等式的定义(有关详细信息，请参阅 [35])。

定义3.1. *(马尔科夫不等式)* 若X为一个非负随机变量，那么对于任意的实数$a>0$，我们有

$$P(X\geq a)\leq\frac{E[X]}{a}.\tag{66}$$

定理3.3. 假设与定理*3.2* 和(57) 中相同的假设成立。然后我们有$f(\tilde{x}_k)-f(x^*)$ 在$k\to\infty$ 时依

概率收敛到0，即对于任何$\varepsilon \geq 0$，我们有

$$\lim_{k\to\infty} P(f(\tilde{x}_k) - f(x^*) \geq \epsilon) = 0. \tag{67}$$

证明：由定理3.2，我们得

$$E[f(\tilde{x}_k) - f(x^*)] \leq \rho^k E[f(\tilde{x}_0) - f(x^*)], \tag{68}$$

因为$\rho < 1$，且$E[f(\tilde{x}_0) - f(x^*)] \leq M_f$, 所以当$k \to \infty$ 时，$E[f(\tilde{x}_k) - f(x^*)] \to 0$. 又由于$f(\tilde{x}_k) - f(x^*)$ 是一个非负随机变量，应用马尔科夫不等式(66), 我们有

$$P(f(\tilde{x}_k) - f(x^*) \geq \epsilon) \leq E[f(\tilde{x}_k) - f(x^*)] \to 0. \tag{69}$$

定理3.3 表明我们的目标函数的值在依概率收敛。

4. 数值实验

在本节中，我们解决了几个流行的有监督的机器学习任务，包括岭回归、逻辑回归和支持向量机问题，将算法STCGVR 与SVRG 进行比较。显然，前两种是光滑且强凸的优化模型，第三种是光滑非凸的优化模型。本文涉及的算法程序都在Matlab R2016a 处理器上运行。

因为STCGVR 和SVRG 需要计算完整的梯度，所以每个epoch 都需要所有的数据。为了降低这些额外成本，我们的图表显示了函数损失值相对于通过数据的次数的关系。也就是说，横轴表示数据的有效通过次数，纵轴表示损失函数的值。epoch 的最大值设置为20。在这两种方法中，我们设置$c1 = 10^{-4}$ 和$c_2 = 0.1$，$M_1 = 1$。

4.1. 岭回归问题

在我们的第一个实验中，我们选择岭回归问题来验证我们的算法。岭回归，也称为Tikhonov正则化，是机器学习模型中本质上至关重要的学习模型。岭回归的目标是最小化成本函数

$$\min_x \frac{1}{n}\sum_{i=1}^{n}(b_i - a_i^T x)^2 + \lambda||x||_2^2, \tag{70}$$

其中$a_i \in R^n$和$b_i \in \{-1, 1\}$ 分别是第i 个例子的特征向量和目标值，$\lambda > 0$ 是正则化参数。

在实验中，我们得到了STCGVR 和SVRG 在求解问题(70) 时的数值结果。我们将两种方法的初始点设置为$x_1 = 5\bar{x}_1$，其中$\bar{x}_1$ 是多维标准正态向量，其具有大约10% 的非零元素。我们以下列方式生成训练集和测试集(a, b)。我们首先生成一个随机向量a,它从$[-0.5, 0.5]^n$ 上的均匀分布中抽取，然后为一些从均匀分布中抽取的$a \in R^n$设置标签$b \in \{-1, 1\}$，$b = sign(\langle\tilde{x}, a\rangle)$在$[-1, 1]$ 上。此外，我们设置正则化参数$\lambda = 10^{-4}$，并选择随机梯度步数$m = n/5$。

图 1比较了SVRG 和STCGVR 的迭代效率，其中横坐标表示全局有效迭代次数，纵坐标表示损失函数值。迭代效率反映了目标函数损失值随迭代次数的增加而呈现出的变化趋势。从图 1可以

看出，对于算法SVRG，我们设置步长$\alpha = 0.005$；对于STCGVR，我们使用Wolfe 线搜索来选择合适的步长。与SVRG 相比，STCGVR 算法只需要大约4 次外循环就可以快速逼近函数的最小损失值10^{-3}，即大约400 次迭代后收敛。结果表明，STCGVR算法在方差缩减的基础上添加了三项共轭梯度，因此能够在极少的迭代次数内逼近最优解。

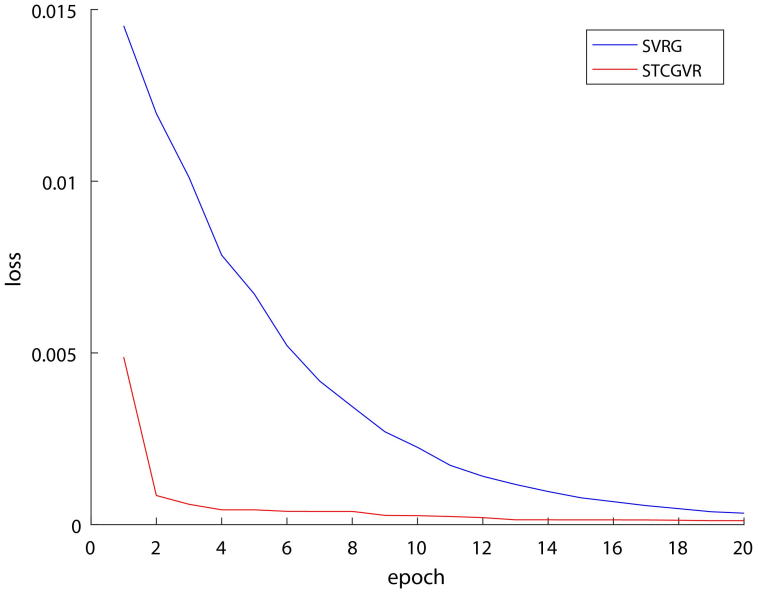

Figure 1. Training loss of SVRG and STCGVR for the ridge regression problem

图 1. SVRG和STCGVR对于岭回归问题的训练损失

4.2. 逻辑回归问题

在第二个实验中，我们考虑ℓ_2逻辑回归(LR) 问题：

$$\min_x \frac{1}{n}\sum_{i=1}^{n} ln(1 + exp(-b_i a_i^T x)) + \lambda ||x||^2, \tag{71}$$

其中$\lambda > 0$ 是正则化参数，$a \in R^n$表示特征向量，$b \in \{-1, 1\}$ 是指相应的标签。

我们令$\lambda = 10^{-4}$，$m = n/5$，固定步长$\alpha = 0.015$，将两种方法的初始点设置为$x_1 = 5\bar{x}_1$，其中$\bar{x}_1$ 是从均匀分布$[0, 1]^n$ 随机抽取的。我们以下列方式生成训练集和测试集(a, b)。我们首先生成一个随机向量a，其中5% 的非零元素从$[0, 1]^n$ 上的均匀分布中抽取，然后为一些从均匀分布中抽取的$a \in R^n$设置标签$b \in \{-1, 1\}$，$b = sign(\langle \tilde{x}, a \rangle)$ 在$[-1, 1]$ 上。

图 2比较了SVRG 和STCGVR 的迭代效率，其中横坐标表示全局有效迭代次数，纵坐标表示损失函数值。迭代效率反映了目标函数损失值随迭代次数的增加而呈现出的变化趋势。随着有效迭代次数的增加，我们可以观察到SVRG 的损失函数的值在大约9 次全梯度计算后逐渐减小并趋于稳定。由于$m = n/5$，也就是说，它在大约900 次迭代后收敛，并且收敛到最优值点的速度很慢。然而，STCGVR只需要300 次迭代即可达到相同的精度。结果表明，在解决逻辑回归问题时，在

每次内循环迭代开始时，三项共轭梯度方向以最陡下降方向重新开始迭代，有效地提高了收敛速度。

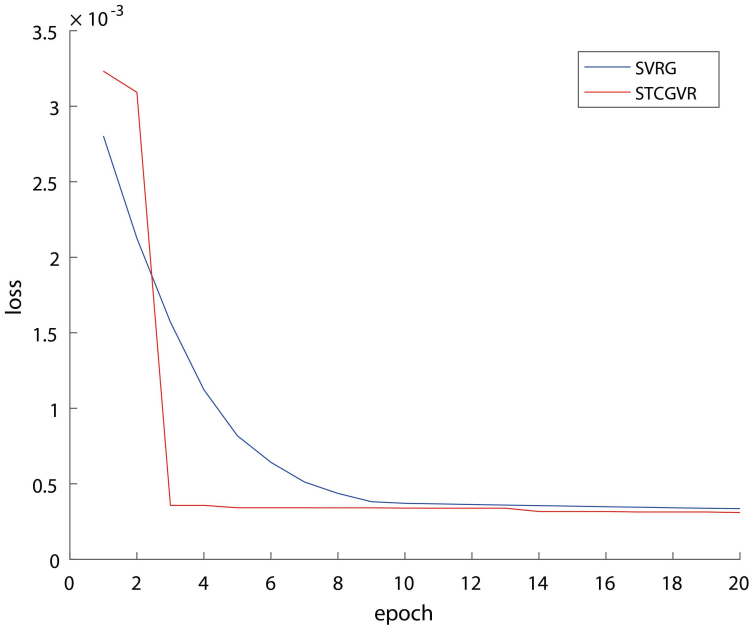

Figure 2. Training loss of SVRG and STCGVR for logistic regression problem

图 2. SVRG和STCGVR对于逻辑回归问题的训练损失

4.3. 非凸支持向量机问题

在我们的最后一个实验中，我们考虑了一个非凸支持向量机(SVM) 问题。给定一个具有已知类的点的训练集，SVM 的目标是找到一个能最好地分离训练集的超平面。我们令$S = \{(a_i, b_i)\}_{i=1}^n$是一个包含n 对形式的训练集$(a_i, b_i)$，其中$a_i \in R^n$是特征向量，$b_i \in \{-1, 1\}$是对应的标签。目标是找到一个由向量$x \in R^n$支持的超平面，该向量将训练集分开，使得对于$b_i = 1$的所有点$x^T a_i > 0$，对于$b_i = -1$ 的所有点$x^T a_i < 0$。如果数据不是完全可分离的，则该向量可能不存在，或者，如果数据是可分离的，则可能有多个分离向量。

我们通过使用sigmoid损失函数解决以下非凸支持向量机(SVM)问题来比较SVRG和STCGVR的收敛性能，这已在 [35]中进行过考虑:

$$\min_{x \in R^n} f(x) = E_{a,b}[1 - tanh(b\langle x, a\rangle)] + \lambda ||x||_2^2, \tag{72}$$

其中$\lambda > 0$是一个正则参数。在我们的实验中，λ 被设置为10^{-4}。在实际情况下，等式(72)写成

$$\min_{x \in R^n} \frac{1}{n} \sum_{i=1}^{n} f_i(x) + \lambda ||x||^2, \tag{73}$$

其中$f_i(x) = 1 - tanh(b_i\langle x, a_i\rangle), i = 1, ..., n.$

我们通过问题(4.73)在合成数据上的表现得到了SVRG 和STCGVR 的数值结果。我们将两种方法的初始点设置为$x_1 = 5\bar{x}_1$，其中$\bar{x}_1$ 是从均匀分布$[0,1]^n$上随机抽取的。我们以下列方式生成训练集和测试集(a,b)。我们首先生成一个具有80% 的非零分量的稀疏向量a，其服从$[0,1]^n$ 上的均匀分布，然后令$b = sign(\langle \bar{x}, a\rangle)$。

图 3比较了SVRG 和STCGVR 的迭代效率，其中横坐标表示全局有效迭代次数，纵坐标表示损失函数值。迭代效率反映了目标函数损失值随迭代次数的增加而呈现出的变化趋势。在不断调整步长后，我们最终选择$\alpha = 5 \times 10^{-3}$。此时，SVRG 在求解非凸SVM 问题时取得了较好的收敛性。从图 3可以看出，我们的方法只需要大约200 次即可收敛。然而，SVRG 需要大约1200 次迭代才能达到相似的精度。实验结果表明，我们提出的方法在求解非凸支持向量机问题时具有更快的收敛速度。

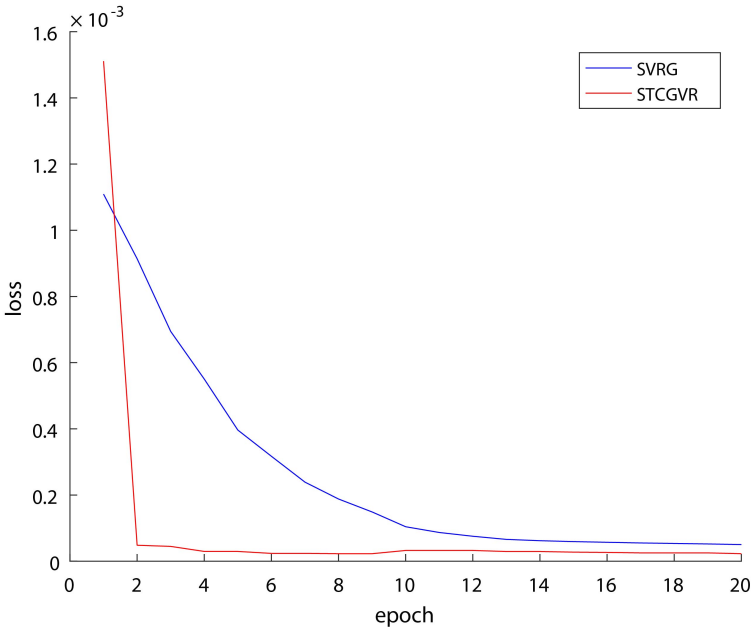

Figure 3. Training loss of SVRG and STCGVR for nonconvex SVM

图 3. SVRG和STCGVR对于非凸SVM的训练损失

5. 总结

本文提出了一种用于解决无约束随机优化问题的STCGVR 算法。这种新颖的算法结合了随机方差减少技术和改进的Dai-Liao 三项共轭梯度，以获得更好的收敛性。此外，在每次内循环迭代开始时都考虑了重新启动技术，这将定期重启算法并擦除对算法可能不利的旧信息。在适当的条件下，有效验证了STCGVR的线性收敛性。我们通过求解几个机器学习问题，这些问题可能是凸的，也可能是非凸的，得到了令人鼓舞的数值结果。在未来的研究中，随着算法的数值梯度实现，STCGVR 可以很容易地应用于不同的实际问题，例如低秩矩阵恢复和稀疏字典学习问题。

基金项目

国家自然科学基金青年基金项目(11601252)。

参考文献

[1] Kawaguchi, K. and Lu, H.H. (2020) Ordered SGD: A New Stochastic Optimization Framework for Empirical Risk Minimization. *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, **108**.

[2] Shalev-Shwartz, S. and Ben-David, S. (2014) References. In: *Understanding Machine Learning*, Cambridge University Press, Cambridge, 385-394. https://doi.org/10.1017/CBO9781107298019.036

[3] Taheri, H., Pedarsani, R. and Thrampoulidis, C. (2021) Fundamental Limits of Ridge-Regularized Empirical Risk Minimization in High Dimensions. *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, **130**.

[4] Shalev-Shwartz, S. and Srebro, N. (2008) SVM Optimization: Inverse Dependence on Training Set Size. *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, 5-9 June 2008. https://doi.org/10.1145/1390156.1390273

[5] Bottou, L. (2010) Large-Scale Machine Learning with Stochastic Gradient Descent. In: Lechevallier, Y. and Saporta, G., Eds., *Proceedings of COMPSTAT'2010*, Physica-Verlag HD, 177-186. https://doi.org/10.1007/978-3-7908-2604-3_16

[6] Bottou, L., Curtis, F.E. and Nocedal, J. (2018) Optimization Methods for Large-Scale Machine Learning. *SIAM Review*, **60**, 223-311. https://doi.org/10.1137/16M1080173

[7] Mokhtari, A. and Ribeiro, A. (2013) A Dual Stochastic DFP Algorithm for Optimal Resource Allocation in Wireless Systems. *2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Darmstadt, 16-19 June 2013, 21-25. https://doi.org/10.1109/SPAWC.2013.6612004

[8] Couillard, O. (2020) Fast and Flexible Optimization of Power Allocation in Wireless Communication Systems Using Neural Networks. McGill University, Montreal, Canada.

[9] Robbins, H. and Monro, S. (1951) A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, **22**, 400-407. https://doi.org/10.1214/aoms/1177729586

[10] Le Roux, N., Schmidt, M. and Bach, F. (2012) A Stochastic Gradient Method with an Exponential Convergence Rate for Finite Training Sets. arXiv preprint arXiv:1202.6258

[11] Schmidt, M., Le Roux, N. and Bach, F. (2017) Minimizing finite Sums with the Stochastic Average Gradient. *Mathematical Programming*, **162**, 83-112. https://doi.org/10.1007/s10107-016-1030-6

[12] Defazio, A., Bach, F. and Lacoste-Julien, S. (2014) SAGA: A Fast Incremental Gradient Method with Support for Non-Strongly Convex Composite Objectives. In: *Advances in Neural Information Processing Systems 27 (NIPS 2014)*.

[13] Kulunchakov, A. (2020) Stochastic Optimization for Large-Scale Machine Learning: Variance Reduction and Acceleration. Grenoble Alpes University, France.

[14] Wang, C., *et al.* (2013) Variance Reduction for Stochastic Gradient Optimization. In: *Advances in Neural Information Processing Systems 26 (NIPS 2013)*.

[15] Shen, Z., *et al.* (2016) Adaptive Variance Reducing for Stochastic Gradient Descent. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, New York, July 2016, 1990-1996.

[16] Konečný, J. and Richtárik, P. (2017) Semi-Stochastic Gradient Descent Methods. *Frontiers in Applied Mathematics and Statistics*, **3**, Article 9. https://doi.org/10.3389/fams.2017.00009

[17] Shang, F., *et al.* (2021) Efficient Asynchronous Semi-Stochastic Block Coordinate Descent Methods for Large-Scale SVD. *IEEE Access*, **9**, 126159-126171. https://doi.org/10.1109/ACCESS.2021.3094282

[18] Duchi, J., Hazan, E. and Singer, Y. (2011) Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, **12**, 2121-2159.

[19] Tieleman, T. and Hinton, G. (2012) Lecture 6.5-rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude. *COURSERA: Neural Networks for Machine Learning*, **4**, 26-31.

[20] Kingma, D.P. and Ba, J. (2014) Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980

[21] Mokhtari, A. and Ribeiro, A. (2013) Regularized Stochastic BFGS Algorithm. *2013 IEEE Global Conference on Signal and Information Processing*, Austin, TX, 3-5 December 2013, 1109-1112. https://doi.org/10.1109/GlobalSIP.2013.6737088

[22] Byrd, R.H., *et al.* (2016) A Stochastic Quasi-Newton Method for Large-Scale Optimization. *SIAM Journal on Optimization*, **26**, 1008-1031. https://doi.org/10.1137/140954362

[23] Liu, D.C. and Nocedal, J. (1989) On the Limited Memory BFGS Method for Large Scale Optimization. *Mathematical Programming*, **45**, 503-528. https://doi.org/10.1007/BF01589116

[24] Moritz, P., Nishihara, R. and Jordan, M. (2016) A Linearly-Convergent Stochastic L-BFGS Algorithm. *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, **41**.

[25] Gower, R., Goldfarb, D. and Richtárik, P. (2016) Stochastic Block BFGS: Squeezing More Curvature Out of Data. *International Conference on Machine Learning*, New York, June 2016.

[26] Fletcher, R. and Reeves, C.M. (1964) Function Minimization by Conjugate Gradients. *The Computer Journal*, **7**, 149-154. https://doi.org/10.1093/comjnl/7.2.149

[27] Andrei, N. (2013) On Three-Term Conjugate Gradient Algorithms for Unconstrained Optimization. *Applied Mathematics and Computation*, **219**, 6316-6327. https://doi.org/10.1016/j.amc.2012.11.097

[28] Yao, S.W., *et al.* (2020) A Class of Globally Convergent Three-Term Dai-Liao Conjugate Gradient Methods. *Applied Numerical Mathematics*, **151**, 354-366. https://doi.org/10.1016/j.apnum.2019.12.026

[29] Dai, Y.-H. and Liao, L.-Z. (2001) New Conjugacy Conditions and Related Nonlinear Conjugate Gradient Methods. *Applied Mathematics and Optimization*, **43**, 87-101. https://doi.org/10.1007/s002450010019

[30] Babaie-Kafaki, S. and Ghanbari, R. (2014) A Descent Family of Dai-Liao Conjugate Gradient Methods. *Optimization Methods and Software*, **29**, 583-591. https://doi.org/10.1080/10556788.2013.833199

[31] Andrei, N. (2015) A New Three-Term Conjugate Gradient Algorithm for Unconstrained Optimization. *Numerical Algorithms*, **68**, 305-321. https://doi.org/10.1007/s11075-014-9845-9

[32] Yao, S.W., *et al.* (2020) A Class of Globally Convergent Three-Term Dai-Liao Conjugate Gradient Methods. *Applied Numerical Mathematics*, **151**, 354-366. https://doi.org/10.1016/j.apnum.2019.12.026

[33] Powell, M.J.D. (1977) Restart Procedures for the Conjugate Gradient Method. *Mathematical Programming*, **12**, 241-254. https://doi.org/10.1007/BF01593790

[34] Jiang, X.z., *et al.* (2021) An Improved Polak-Ribière-Polyak Conjugate Gradient Method with an Efficient Restart Direction. *Computational and Applied Mathematics*, **40**, Article No. 174. https://doi.org/10.1007/s40314-021-01557-9

[35] Zoutendijk, G. (1966) Nonlinear Programming: A Numerical Survey. *SIAM Journal on Control*, **4**, 194-210. https://doi.org/10.1137/0304019