International Journal of Recent Engineering Science (IJRES),
ISSN: 2349-7157, *Volume 4 Issue 4 July to August 2017*

Improved Stochastic Gradient Descent Algorithm for SVM

Shuxia Lu, Zhao Jin
(Key Lab. of Machine Learning and Computational Intelligence,
College of Mathematics and Information Science, Hebei University, China)

Abstract

In order to improve the efficiency and classification ability of Support vector machines (SVM) based on stochastic gradient descent algorithm, three algorithms of improved stochastic gradient descent (SGD) are used to solve support vector machine, which are Momentum, Nesterov accelerated gradient (NAG), RMSprop. The experimental results show that the algorithm based on RMSprop for solving the linear support vector machine has faster convergence speed and higher testing precision on five datasets (Alpha, Gamma, Delta, Mnist, Usps).

Keywords - *Stochastic gradient descent, Support vector machines, Momentum, Nesterov accelerated gradient, RMSprop.*

I. INTRODUCTION

Stochastic gradient descent (SGD) is a simple and effective method, many works focus on designing variants of SGD. There are some methods that solve SVM problem by using variants of SGD. Some popular methods include the Pegasos method[1], Pegasos performed stochastic gradient descent on the primal objective with a carefully chosen step size, which improves and guarantees convergence. There are some methods for SVM that are proven to converge linearly on strong convex problems. Such as the stochastic gradient descent with Barzilai-Borwein update step for SVM[2], Budgeted Stochastic Gradient Descent for Large-Scale SVM Training[3], Bi-level stochastic gradient for large-scale support vector machine[4], and the stochastic variance reduced gradient method[5].

Some recent works that discuss the improved approaches for SGD[6-12], such as quasi-Newton stochastic gradient descent, accelerated proximal stochastic dual coordinate ascent, stochastic dual coordinate ascent methods, scalability of stochastic gradient descent based on smart sampling techniques, and beyond the regret barrier algorithms for stochastic strongly convex optimization.

In this paper, we focus on the problem of improving the efficiency and classification ability of Support vector machines (SVM) based on stochastic gradient descent algorithm, three algorithms of improved stochastic gradient descent (SGD) are used to solve support vector machine, which are Momentum, Nesterov accelerated gradient (NAG), RMSprop. The experimental results show that the algorithm based on RMSprop for solving the linear support vector machine has faster convergence speed and higher testing precision on five datasets (Alpha, Gamma, Delta, Mnist, Usps).

II. STOCHASTIC GRADIENT DESCENT FOR SVM

In order to deal with the large-scale data classification problems, we describe the algorithms of stochastic gradient descent for SVM.

Consider a binary classification problem with examples $S=\{(\mathbf{x}_i, y_i),\ \ i=1,\cdots,N\}$, where instance $\mathbf{x}_i \in R^d$ is a d-dimensional input vector and $y_i \in \{+1,-1\}$ is the label. Training an SVM classifier $f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}^T\mathbf{x})$ using S, where $\mathbf{w}$ is a vector of weights associated with each input, which is formulated as solving the following optimization problem

$$\min\ \ p_t(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + l(\mathbf{w};(\mathbf{x}_t, y_t))\ , \qquad (1)$$

where $l(\mathbf{w};(\mathbf{x}_t, y_t)) = \max(0, 1 - y_t\mathbf{w}^T\mathbf{x}_t)$ is the *hinge loss* function and $\lambda \ge 0$ is a regularization parameter used to control model complexity.

SGD works iteratively. It starts with an initial guess of the model weight $\mathbf{w}_1$, and at t-th round it updates the current weight $\mathbf{w}_t$ as

$$\begin{aligned}\mathbf{w}_{t+1} &= \mathbf{w}_t - \eta_t \nabla_t p_t(\mathbf{w}_t) \\ &= (1-\eta_t\lambda)\mathbf{w}_t + \eta_t \mathbf{1}[\mathrm{y}_t \langle \mathbf{w}_t, \mathbf{x}_t \rangle]\,\mathrm{y}_t\,\mathbf{x}_t\end{aligned} \qquad (2)$$

where

$$\mathbf{1}[\mathrm{y}_t \langle \mathbf{w}_t, \mathbf{x}_t \rangle < 1] = \begin{cases} 1, & \textit{if}\ \mathrm{y}_t \langle \mathbf{w}_t, \mathbf{x}_t \rangle < 1 \\ 0, & \textit{otherwise.} \end{cases}$$

which is the indicator function which takes a value of one if its argument is true (**w** yields non-zero loss on the example (**x**, y)), and zero otherwise. We then update using a step size of $\eta_t = 1/(\lambda \mathrm{t})$. After a predetermined number T of iterations, we output the last iterate $\mathbf{w}_{t+1}$.

Then, the decision function for SVM with SGD is as follows

$$f_{t+1}(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}_{t+1}{}^{T}\mathbf{x}) \tag{3}$$

III. IMPROVED STOCHASTIC GRADIENT DESCENT ALGORITHM FOR SVM

Stochastic gradient descent parameter update rule:

$$\theta = \theta - \eta \cdot \nabla_{\theta} J\left(\theta; x^{(i)}, y^{(i)}\right) \tag{4}$$

In the following, we present three adaptive learning rate SGD algorithms for SVM. It is especially suited for learning from large datasets.

A. Momentum SVM

Momentum [13] is a method that helps accelerate SGD in the relevant direction and dampens oscillations. We use Momentum method to optimize SVM.

The Momentum SVM update rule:

$$\begin{aligned} v_{t+1} &= \gamma v_t + \eta\left(\lambda \mathrm{w}_t - \alpha_t y_{i_t} x_{i_t}\right) \\ \mathrm{w}_{t+1} &= \mathrm{w}_t - v_{t+1} \end{aligned} \tag{5}$$

Algorithm 1：Momentum SVM

```
Input： $S$，λ， $T$, $\gamma$,  $\eta$
Initialize： $\mathrm{w}_1 = \vec{0}$ ， $v_1 = \vec{0}$,  $\gamma = 0.9$
for $t = 1, \cdots, T$
    choose $i_t \in \{ 1, \ldots, |\mathrm{S}| \}$ uniformly at random
    if $y_{i_t} \langle \mathrm{w}_t, x_{i_t} \rangle < 1$, then
        $v_{t+1} = \gamma v_t + \eta\left(\lambda \mathrm{w}_t - y_{i_t} x_{i_t}\right)$
    else
        $v_{t+1} = \gamma v_t + \eta \lambda w_t$
    $\mathrm{w}_{t+1} = \mathrm{w}_t - v_{t+1}$
Output： $\mathrm{w}_{T+1}$
```

B. RMSprop SVM

RMSprop [14] is an adaptive learning rate method proposed by Geoff Hinton.

The RMSprop update rule:

$$\begin{aligned} E[\nabla^2]_t &= 0.9E[\nabla^2]_{t-1} + 0.1\nabla_{\theta} J(\theta_t)^2 \\ \theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{E[\nabla^2]_t + \varepsilon}} \nabla_{\theta} J(\theta_t) \end{aligned} \tag{6}$$

Algorithm 2：RMSprop SVM

```
Input： $S$，λ， $T$,  $\varepsilon$,  $\gamma$,  $\eta$
Initialize： $w_1 = \vec{0}$ ， $E[\nabla^2]_1 = \vec{0}$,  $\varepsilon = 1e-8$,
    $\gamma = 0.9$,  $\eta = 0.01$
for $t = 1, \cdots, T$
    choose $i_t \in \{ 1, \ldots, |\mathrm{S}| \}$ uniformly at random
    if $y_{i_t} \langle \mathrm{w}_t, x_{i_t} \rangle < 1$, then
        $\nabla_{t+1} = \lambda \mathrm{w}_t - \alpha_t y_{i_t} x_{i_t}$
    else
        $\nabla_{t+1} = \lambda \mathrm{w}_t$
    $E[\nabla^2]_{t+1} = \gamma E[\nabla^2]_t + (1-\gamma)\nabla_{t+1}{}^2$
    $\mathrm{w}_{t+1} = \mathrm{w}_t - \frac{\eta}{E[\nabla^2]_{t+1}} \otimes \nabla_{t+1}$
Output： $\mathrm{w}_{T+1}$
```

C. NAG SVM

Nesterov accelerated gradient (NAG) [15] is a way to look ahead by calculating the gradient not w.r.t. to our current parameters but w.r.t. the approximate future position of our parameters. The parameter update takes the form:

$$\begin{aligned} v_{t+1} &= \gamma v_t + \eta\left(\lambda\left(\mathrm{w}_t - \gamma v_t\right) - \alpha_t y_{i_t} x_{i_t}\right) \\ \mathrm{w}_{t+1} &= \mathrm{w}_t - v_{t+1} \end{aligned} \tag{7}$$

Algorithm 3：NAG SVM

```
Input： $S$，λ， $T$,  $\gamma$,  $\eta$
Initialize： $\mathrm{w}_1 = \vec{0}$ ， $v_1 = \vec{0}$,  $\gamma = 0.9$
for $t = 1, \cdots, T$
    choose $i_t \in \{ 1, \ldots, |\mathrm{S}| \}$ uniformly at random
    if $y_{i_t} \langle \mathrm{w}_t, x_{i_t} \rangle < 1$, then
        $v_{t+1} = \gamma v_t + \eta\left(\lambda\left(\mathrm{w}_t - \gamma v_t\right) - y_{i_t} x_{i_t}\right)$
    else
        $v_{t+1} = \gamma v_t + \eta \lambda\left(\mathrm{w}_t - \gamma v_t\right)$
    $\mathrm{w}_{t+1} = \mathrm{w}_t - v_{t+1}$
Output： $\mathrm{w}_{T+1}$
```

IV. EXPERIMENTAL RESULTS

In this section, we perform some experiments that demonstrate the efficacy of our algorithm. The basic SGD algorithm is Pegasos [1]. To evaluate the classification accuracy and convergence rate of four methods, several datasets are used to illustrate in the linear kernel situations. Machine has four E5-2609 2.50GHz processors and 4GB RAM memory.

We tested the performance of four methods on three large datasets and three standard real datasets, three large datasets are derived from Pascal Large Scale Learning Challenge, and three standard real datasets are downloaded from LIBSVM website. The Usps and Mnist datasets are used for the task of classifying digits 0, 1, 2, 3, 4 versus the rest of the classes. The original Letter dataset’s labels represent 26 alphabets and we set the former 13 alphabets as positive class and the rest as negative class. We use the linear kernel and the regularization parameter λ in our experiments. The datasets characteristics are given in Table 1.

Table 1 Datasets and Parameters

Dataset	#Training	#Testing	#Features
Alpha	400,000	100,000	500
Gamma	400,000	100,000	500
Delta	400,000	100,000	500
Mnist	60,000	10,000	780
Letter	15,000	5,000	16
Usps	7,291	2,007	256

Table 2 shows the testing accuracy of four methods for linear kernel on six datasets.

Table 2 Comparisons of Four Methods

Dataset	Pegasos	Momentum	RMSprop	NAG
Alpha	72.68	71.56	75.05	71.82
Gamma	73.15	70.82	77.04	70.71
Delta	70.77	65.55	74.60	65.81
Mnist	87.03	83.82	84.78	84.17
Letter	73.51	73.16	73.74	73.59
Usps	83.83	82.46	83.64	82.23

Figures 1-6 shows the convergence rate four methods with the number of iteration growing.

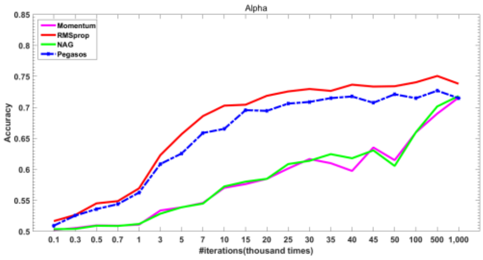

Fig 1 Comparisons of Four Methods on Alpha Dataset

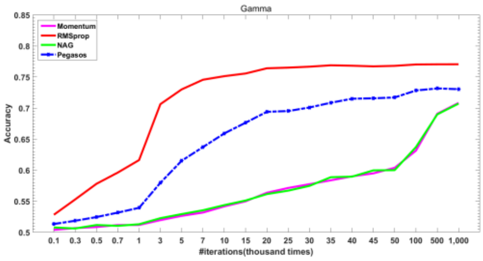

Fig 2 Comparisons of Four Methods on Gamma Dataset

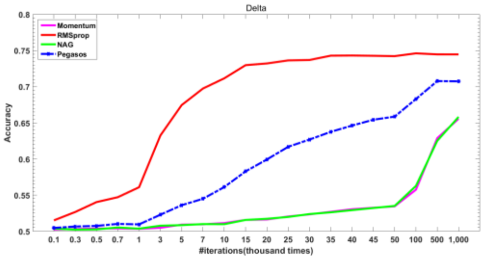

Fig 3 Comparisons of Four Methods on Delta Dataset

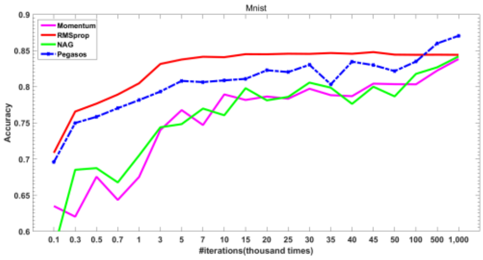

Fig 4 Comparisons of Four Methods on Mnist Dataset

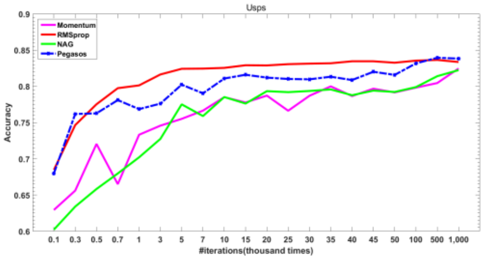

Fig 5 Comparisons of Four Methods on Usps Dataset

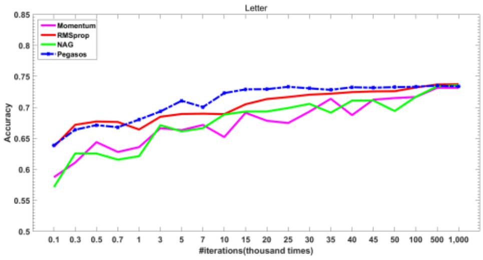

Fig 6 Comparisons of Four Methods on Letter Dataset

Figures 1-5 shows that RMSprop SVM method for linear kernel has a faster convergence rate than other methods on five datasets (Alpha, Gamma, Delta, Mnist, Usps). Figure 6 show that Pegasos has a faster convergence rate than other methods on Letter dataset.

V. CONCLUSION

In this paper, we focus on the problem of improving the efficiency and classification ability of Support vector machines (SVM) based on stochastic gradient descent algorithm, three algorithms of improved stochastic gradient descent (SGD) are used to solve support vector machine, which are Momentum, Nesterov accelerated gradient (NAG), RMSprop. The experimental results show that the algorithm based on RMSprop for solving the linear support vector machine has faster convergence speed and higher testing precision on five datasets (Alpha, Gamma, Delta, Mnist, Usps). Pegasos has a faster convergence rate than other methods on Letter dataset.

ACKNOWLEDGEMENTS

This research is supported by the natural science foundation of Hebei Province No. F2015201185.

REFERENCES

[1] Shalev-Shwartz, Y. Singer, N. Srebro, et al, Pegasos: Primal Estimated sub-Gradient Solver for SVM, Mathematical Programming, 127(1), 2011, 3-30.

[2] Krzysztof Sopyla, Pawel Drozda, Stochastic Gradient Descent with Barzilai-Borwein update step for SVM, Information Sciences, 316, 2015, 218-233.

[3] Zhuang Wang, Koby Crammer, Slobodan Vucetic, Breaking the Curse of Kernelization: Budgeted Stochastic Gradient Descent for Large-Scale SVM Training, Journal of Machine Learning Research, 13, 2013, 3103-3131.

[4] Nicolas Couellan, Wenjuan Wang, Bi-level stochastic gradient for large scale support vector machine, Neurocomputing, 153, 2015, 300-308.

[5] R. Johnson and T. Zhang, Accelerating Stochastic Gradient Descent using predictive variance reduction. In Advances in Neural Information Processing Systems, 2013, 315-323.

[6] A. Bordes, L. Bottou, P. Gallinari, SGD-QN: careful quasi-Newton stochastic gradient descent, J. Mach. Learn, 10, 2009, 1737-1754.

[7] A. Bordes, L. Bottou, P. Gallinari, et al, Sgdqn is less careful than expected, J. Mach. Learn, 11, 2010, 2229-2240.

[8] Shai Shalev-Shwartz, Tong Zhang, Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization, Math. Program., 155, 2016, 105-145.

[9] Shalev-Shwartz, Zhang, et al, Stochastic dual coordinate ascent methods for regularized losss minimization, J. Mach. Learn, 14, 2013, 567-599.

[10] Stephan Clemencon, Aurelien Bellet, Ons Jelassi, et al, Scalability of Stochastic Gradient Descent based on Smart Sampling Techniques, Procedia Computer Science, 53, 2015, 308–315.

[11] Elad Hazan, Satyen Kale, Beyond the Regret Minimization Barrier: Optimal Algorithms for Stochastic Strongly Convex Optimization, Journal of Machine Learning Research, 15, 2014, 2489-2512.

[12] Z. Lei, Y. Yang, Z. Wu, Ensemble of support vector machine for text-independent speaker recognition, International Journal Computer Science and Network Security, 6 (1), 2006, 163–167.

[13] Ning Qian, On the momentum term in gradient descent learning algorithms, Neural networks: the official journal of the International Neural Network Society, 12(1), 1999, 145–151.

[14] Arvind Neelakantan, Luke Vilnis, Quoc V. Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens, Adding Gradient Noise Improves Learning for Very Deep Networks, 2015, 1-11.

[15] Yurii Nesterov, A method for unconstrained convex minimization problem with the rate of convergence o(1/k2). Doklady ANSSSR (translated as Soviet. Math. Docl.), 269, 543–547.

[16] S. Sonnenburg, V. Franc, E.Y. Tov, M. Sebag, PASCAL large-scale learning challenge, 2008.

[17] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2016.