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

\[
\min_p \, p(w) = \frac{1}{2} w^T w + l(w; (x_i, y_i)), \tag{1}
\]

where \(l(w; (x_i, y_i)) = \max(0, 1 - y_i w^T x_i)\) is the hinge loss function and \(\lambda \geq 0\) is a regularization parameter used to control model complexity.

SGD works iteratively. It starts with an initial guess of the model weight \(w_0\), and at \(t\)-th round it updates the current weight \(w_t\) as

\[
w_{t+1} = w_t - \eta_t \nabla_p l(w_t, (x_i, y_i)) = (1 - \eta_t \lambda)w_t + \eta_t l[y_t, \langle w_t, x_i \rangle]y_i, x_i \tag{2}
\]

where

\[
l[y_t, \langle w_t, x_i \rangle] < 1 = \begin{cases} 1, & \text{if } y_t \langle w_t, x_i \rangle < 1 \\ 0, & \text{otherwise.} \end{cases}
\]

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 \(\eta_t = 1/(\lambda t)\). After a predetermined number \(T\) of iterations, we output the last iterate \(w_{t+1}\).

Then, the decision function for SVM with SGD is as follows.
\[ f_{i,t}(x) = \text{sgn}(w_{i,t}^T x) \]  

(3)

III. IMPROVED STOCHASTIC GRADIENT DESCENT ALGORITHM FOR SVM

Stochastic gradient descent parameter update rule:

\[ \theta = \theta - \eta \cdot \nabla_o J \left( \theta; x^{(i)}; y^{(i)} \right) \]  

(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:

\[ v_{i+1} = \gamma v_i + \eta \left( \lambda w_i - \alpha y_i x_i \right) \]

\[ w_{i+1} = w_i - v_{i+1} \]  

(5)

Algorithm 1: Momentum SVM

1. Input: \( S, \lambda, T, \gamma, \eta \)
2. Initialize: \( w_i = \tilde{0}, v_i = \tilde{0}, \gamma = 0.9 \)
3. for \( t = 1, \ldots, T \)
4. \( i \in \{ 1, \ldots, |S| \} \) uniformly at random
5. if \( y_i \left( w_i, x_i \right) < 1 \), then
6. \( v_{i+1} = \gamma v_i + \eta \left( \lambda w_i - y_i x_i \right) \)
7. else
8. \( v_{i+1} = \gamma v_i + \eta \lambda w_i \)
9. \( w_{i+1} = w_i - v_{i+1} \)
10. Output: \( w_{T+1} \)

B. RMSprop SVM

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

The RMSprop update rule:

\[ E[\nabla^2] = 0.9 E[\nabla^2], \quad \theta_{i+1} = \theta_i - \eta \frac{\nabla_o J(\theta_i)}{\sqrt{E[\nabla^2] + \epsilon}} \]  

(6)

Algorithm 2: RMSprop SVM

1. Input: \( S, \lambda, T, \epsilon, \gamma, \eta \)
2. Initialize: \( w_i = \tilde{0}, E[\nabla^2] = \tilde{0}, \epsilon = 1e-8, \gamma = 0.9, \eta = 0.01 \)
3. for \( t = 1, \ldots, T \)
4. \( i \in \{ 1, \ldots, |S| \} \) uniformly at random
5. if \( y_i \left( w_i, x_i \right) < 1 \), then
6. \( \nabla_{i+1} = \lambda w_i - \alpha y_i x_i \)

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:

\[ v_{i+1} = \gamma v_i + \eta \left( \lambda w_i - \alpha y_i x_i \right) \]

\[ w_{i+1} = w_i - v_{i+1} \]

(7)

Algorithm 3: NAG SVM

1. Input: \( S, \lambda, T, \gamma, \eta \)
2. Initialize: \( w_i = \tilde{0}, v_i = \tilde{0}, \gamma = 0.9 \)
3. for \( t = 1, \ldots, T \)
4. \( i \in \{ 1, \ldots, |S| \} \) uniformly at random
5. if \( y_i \left( w_i, x_i \right) < 1 \), then
6. \( v_{i+1} = \gamma v_i + \eta \left( \lambda w_i - \gamma v_i \right) - \alpha y_i x_i \)
7. else
8. \( v_{i+1} = \gamma v_i + \eta \lambda (w_i - \gamma v_i) \)
9. \( w_{i+1} = w_i - v_{i+1} \)
10. Output: \( w_{T+1} \)

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 \( \lambda \) in our experiments. The datasets characteristics are given in Table 1.
Table 1 Datasets and Parameters

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Training</th>
<th>#Testing</th>
<th>#Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>400,000</td>
<td>100,000</td>
<td>500</td>
</tr>
<tr>
<td>Gamma</td>
<td>400,000</td>
<td>100,000</td>
<td>500</td>
</tr>
<tr>
<td>Delta</td>
<td>400,000</td>
<td>100,000</td>
<td>500</td>
</tr>
<tr>
<td>Mnist</td>
<td>60,000</td>
<td>10,000</td>
<td>780</td>
</tr>
<tr>
<td>Letter</td>
<td>15,000</td>
<td>5,000</td>
<td>16</td>
</tr>
<tr>
<td>Usps</td>
<td>7,291</td>
<td>2,007</td>
<td>256</td>
</tr>
</tbody>
</table>

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

Table 2 Comparisons of Four Methods

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Pegasos</th>
<th>Momentum</th>
<th>RMSprop</th>
<th>NAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>72.68</td>
<td>71.56</td>
<td>75.05</td>
<td>71.82</td>
</tr>
<tr>
<td>Gamma</td>
<td>73.15</td>
<td>70.28</td>
<td>77.04</td>
<td>70.71</td>
</tr>
<tr>
<td>Delta</td>
<td>70.77</td>
<td>65.55</td>
<td>74.60</td>
<td>65.81</td>
</tr>
<tr>
<td>Mnist</td>
<td>87.03</td>
<td>83.82</td>
<td>84.78</td>
<td>84.17</td>
</tr>
<tr>
<td>Letter</td>
<td>73.51</td>
<td>73.16</td>
<td>73.74</td>
<td>73.59</td>
</tr>
<tr>
<td>Usps</td>
<td>83.83</td>
<td>82.46</td>
<td>83.64</td>
<td>82.23</td>
</tr>
</tbody>
</table>

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 shows 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


