Stochastic Gradient Descent with SVM for Imbalanced Data Classification

Lu Shuxia1*, Zhu Chenxu2, Zhou Mi1

1Key Lab. of Machine Learning and Computational Intelligence, College of Mathematics and Information Science, Hebei University, Baoding City, Hebei Province, 071002, China
2College of Science, Northwest Agriculture & Forestry University, Yangling City, Shanxi Province, 712100, China

*Corresponding Author:
Lu Shuxia
Email: cmclusx@126.com

Abstract: Stochastic Gradient Descent (SGD) is an attractive choice for SVM training. SGD leads to a result that the probability of choosing majority class is far greater than that of minority class for imbalanced classification problem. In order to deal with the large-scale imbalanced data classification problems, a method named stochastic gradient descent algorithm with SVM for imbalanced data classification is proposed. First, to deal with imbalanced data classification problems, we define the weight according to the size of the positive and negative dataset. Then, a fast learning algorithm on large datasets called the weighted stochastic gradient descent algorithm with SVM is proposed, which helps to reduce the hyperplane offset to the minority class, thus solve the large-scale imbalanced data classification problems. Experimental results on real datasets show that the proposed method is effective.

Keywords: Stochastic gradient descent, Weight; Imbalanced data, Large-scale learning, Support vector machines

INTRODUCTION

Recent advances in large-scale learning resulted in many algorithms for training SVMs using large data. CVM [1, 2] and parallel SVMs are the successful methods to train SVM from large data. SGD is a recently popularized approach that can be used for online training of SVM, in which the solver is based on the SGD algorithm and have demonstrated its effectiveness for the classification of large datasets with fast convergence and small memory requirements. Pegasos [5] performed stochastic gradient descent on the primal objective with a carefully chosen step size, which improves and guarantees convergence. Krzysztof [3] proposed stochastic gradient descent with Barzilai-Borwein update step for SVM. Wang [4] gives budgeted stochastic gradient descent for large-scale SVM training. This is achieved by controlling the number of SVs through one of the several budget maintenance strategies. Nicolas [5] proposed a bi-level stochastic gradient for large-scale support vector machine with automatic selection of the hyperparameter. Recently there are many improved approaches for SGD [6-12], such as quasi-Newton stochastic gradient descent, accelerated proximal stochastic dual coordinate ascent, stochastic dual coordinate ascent methods, SGD based on smart sampling techniques.

In the real world, these training samples are not always balanced. In an imbalanced dataset, the majority class have a large percentage for all the samples, while the samples in minority class just occupy a small part of all the samples. Many researchers have worked to solve this problem so that the classification performance of the majority class and that of minority class are good at the same time. To solve this problem, two kinds of methods have been proposed: one is based on sampling method and the other one is based on sample weighting method [13-16]. Sampling method includes: under sampling method and oversampling method. The sample weighting approach to the imbalanced data classification problem is to apply the weights to the training data points.

In this paper, we focus on the large and imbalanced datasets effective classification problem, a stochastic gradient descent algorithm with SVM for imbalanced data classification is proposed. It consists of two stages. The first stage is to obtain the weight according to the size of positive and negative dataset. In the second stage, a fast learning algorithm on large datasets called the weighted stochastic gradient descent algorithm with SVM (WSGD) is proposed, which helps to reduce the hyperplane offset to the minority class, thus solve the large scale imbalanced data classification problems. Experiments on large classification datasets also demonstrated that the proposed method has comparable performance.

WEIGHTED STOCHASTIC GRADIENT DESCENT FOR SVM

In order to deal with the large scale imbalanced data classification problems, we describe the algorithms of weighted stochastic gradient descent for SVM.
The weighted linear stochastic gradient descent for SVM (WLSGD)

Consider a binary classification problem with examples \( S = \{ (x_i, y_i) \mid i = 1, \ldots, N \} \), 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

\[
\min_p p_l(w) = \frac{1}{2} \|w\|^2 + s \cdot \ell(w; (x_i, y_i)),
\]

where \( \ell(w; (x_i, y_i)) = \max(0, 1 - y_i w^T x_i) \) is the hinge loss function and \( \lambda \geq 0 \) is a regularization parameter used to control model complexity. \( s \) is the weight, which is set according to the size of positive and negative dataset. See the section of setting the weight for imbalanced problem in detail.

SGD works iteratively. It starts with an initial guess of the model weight \( w_1 \), and at \( t \)-th round it updates the current weight \( w_t \) as

\[
w_{t+1} = w_t - \eta_t \nabla_p p_l(w_t) = (1 - \eta_t \lambda)w_t + \eta_t s \cdot \mathcal{I}[y_t \langle w_t, x_t \rangle < 1] y_t x_t,
\]

where

\[
\mathcal{I}[y_t \langle w_t, x_t \rangle < 1] = \begin{cases} 
1, & \text{if } y_t \langle w_t, x_t \rangle < 1 \\
0, & \text{otherwise}
\end{cases}
\]

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 WLSGD is as follows

\[
f_{t+1}(x) = \text{sgn}(w_{t+1}^T x)
\]

Setting the Weight for Imbalanced Problem

To deal with imbalanced dataset, we simply set the weight according to the size of positive and negative dataset. The data in the majority class have to receive lower weight than those in the minority class receives.

When the size of positive dataset is \( N_{pos} \) and that of negative dataset is \( N_{neg} \), the weights are defined as

\[
s_t = \begin{cases} 
\frac{1}{N_{pos}} & \text{if } y_t = 1, \\
\frac{1}{N_{neg}} & \text{otherwise}.
\end{cases}
\]

To maintain the weight ratio and make the convergence speed faster, we also use the following weighting formulation

\[
s_t = \begin{cases} 
1 & \text{if } y_t = 1, N_{pos} \geq N_{neg}, \\
\frac{N_{neg}}{N_{pos}} & \text{if } y_t = 1, N_{pos} < N_{neg}, \\
\frac{N_{pos}}{N_{neg}} & \text{if } y_t = -1, N_{pos} \geq N_{neg}, \\
1 & \text{if } y_t = -1, N_{pos} < N_{neg}.
\end{cases}
\]

The weighted linear stochastic gradient descent for SVM (WLSGD) is given in algorithm 1.

**Algorithm 1 WLSGD**

1. Input: data \( S \), regularization parameter \( \lambda \), a predetermined number \( T \) of iterations;
2. Initialize: \( w_1 = \mathbf{0} \);
3. Compute the weight \( s_t \) according to the formulation (5);
4. for \( t = 1, \ldots, T \) do
5. choose \( (x_t, y_t) \) uniformly at random;
6. \( w_{t+1} \leftarrow (1 - \eta_t \lambda)w_t \)
7. if \( y_t \langle w_t, x_t \rangle < 1 \) then
8. \( w_{t+1} \leftarrow w_{t+1} + s_t \eta_t y_t x_t \); // compute \( w_{t+1} \) according to the formulation (2)

Available Online: [http://saspjournals.com/sjpms](http://saspjournals.com/sjpms)

9. else
10. \( w_{i+1} \leftarrow w_{i+1} \);
11. end if
12. end if
13. Output: \( f_{i+1}(x) = \text{sgn}(w_{i+1}^T x) \).

The weighted kernelized stochastic gradient descent for SVM (WKSGD)

SVM for SVM can be used to solve non-linear problems when combined with Mercer kernels. After introducing a nonlinear function \( \phi \) that maps \( x \) from the input to the feature space and replacing \( x \) with \( \phi(x) \), the optimization problem can be described as

\[
\min_{w} p_t(w) = \frac{\lambda}{2} \|w\|_2^2 + s_t \cdot I\left(w; (\phi(x_i), y_i)\right)
\]

(6)

where \( I\left(w; (\phi(x_i), y_i)\right) = \max\left(0, 1-y_i w^T \phi(x_i)\right) \) is the hinge loss function.

At \( t \)-th round it updates the current weight \( w_t \) as

\[
w_{i+1} = w_t - \eta \nabla_{w_t} p_t(w_t); (\phi(x_i), y_i) = (1-\eta \lambda) w_t + \eta \nabla_{w_t} p_t(w_t)
\]

It starts with an initial \( w_0 = 0 \), update a step size \( \eta_t = 1/(\lambda t) \), and for all \( t \) we can rewrite \( w_{i+1} \) as

\[
w_{i+1} = \frac{1}{\lambda t} \sum_{j=1}^{t} \eta \nabla_{w_t} p_t(w_t; (\phi(x_i), y_i)) < 1 \mid y_i \phi(x_i)
\]

(7)

For each \( t \), let \( \alpha_{t,i} \in R^N \) be the vector such that \( \alpha_{t,i} \) counts how many times example \( j \) has been selected so far and we had a non-zero loss on it, namely,

\[
\alpha_{t,i} = \left\{ t \mid t : i = j \land \left( y_j \left(w_t; \phi(x_j)\right) < 1\right) \right\}
\]

(8)

Then, the decision function for WKSGD is as follows:

\[
f_{i+1}(x) = \text{sgn}(w_{i+1}^T \phi(x)) = \frac{1}{\lambda t} \sum_{j=1}^{N} \alpha_{i,t} y_j k(x_j, x)
\]

(9)

Only one element of \( \alpha \) is changed at each iteration. The algorithm does not refer to the implicit mapping \( \phi(\cdot) \) and only use the kernel function. This WKSGD implementation is given in algorithm 2.

Algorithm 2 WKSGD

1. Input: data \( S \), regularization parameter \( \lambda \), a predetermined number \( T \) of iterations
2. Initialize: \( \alpha_0 = 0 \)
3. Compute the weight \( s_t \) according to the formulation (5)
4. for \( t = 1, \ldots, T \) do
5. choose \((x_i, y_i)\) uniformly at random;
6. For all \( j \neq i \), set \( \alpha_{i,t} = \alpha_t[i] \)
7. if \( y_j \frac{1}{\lambda t} \sum_{j=1}^{N} s_{j} \alpha_{i,t} y_j k(x_j, x_i) < 1 \) then
8. \( \alpha_{i,t} = \alpha_{i,t} + 1 \); // count the selected times of example \( j \) with a non-zero loss on it
9. else
10. \( \alpha_{i,t} = \alpha_{i,t} \)
11. end if
12. end if
13. Output: \( f_{i+1}(x) = \text{sgn}(w_{i+1}^T \phi(x)) = \frac{1}{\lambda t} \sum_{j=1}^{N} s_{j} \alpha_{i,t} y_j k(x_j, x) \).
EXPERIMENTAL RESULTS

In this section, we conduct the performance comparison of the four methods for real problems: MNIST, Ijcnnl, Shuttle, Letter, Usps, Adult. Most of the datasets are taken from the UCI machine learning repository [17]. Usps is taken from database [18]. The multi-classification dataset are artificially divided into binary classification dataset, which constitute the imbalanced dataset. The description of datasets is shown in Table 1.

The Gaussian function is taken as the kernel function \( k(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / \sigma^2) \). Set the kernel function width \( \sigma=1.5 \), a predetermined number of iterations \( T=10^6 \). The regularization parameters are shown in Table 2.

### Table 1: Introduction to datasets

<table>
<thead>
<tr>
<th>datasets</th>
<th>#classes</th>
<th>Training size</th>
<th>Testing size ((N_{pos}, N_{neg}))</th>
<th>#features</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>10</td>
<td>60000</td>
<td>10000 ((974, 9026))</td>
<td>780</td>
</tr>
<tr>
<td>Ijcnnl</td>
<td>2</td>
<td>91701</td>
<td>49990 ((4853, 45137))</td>
<td>22</td>
</tr>
<tr>
<td>Shuttle</td>
<td>7</td>
<td>43500</td>
<td>14500 ((2155, 12345))</td>
<td>9</td>
</tr>
<tr>
<td>Letter</td>
<td>26</td>
<td>10000</td>
<td>10000 ((353, 9647))</td>
<td>16</td>
</tr>
<tr>
<td>Usps</td>
<td>10</td>
<td>7291</td>
<td>2007 ((198, 1809))</td>
<td>256</td>
</tr>
<tr>
<td>Adult</td>
<td>2</td>
<td>24974</td>
<td>12554 ((119, 12435))</td>
<td>123</td>
</tr>
</tbody>
</table>

### Table 2: Parameter setting

<table>
<thead>
<tr>
<th>(\lambda)</th>
<th>LSGD (10^4)</th>
<th>WLSGD (10^4)</th>
<th>KSGD (10^4)</th>
<th>WKSGD (10^4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Mnist), (10^4) (Ijcnn), (10^4) (Shuttle), (10^4) (Letter), (10^4) (Usps), (10^4) (Adult)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Considering the imbalanced nature of the training datasets, the geometric mean accuracy is adopted to evaluate the performance of our algorithms,

\[
g = \sqrt{a' \cdot a} \times 100% \]

where

\[
a' = \frac{\text{#positive samples correctly classified}}{\text{# total positive samples classified}} \times 100% ,
\]

\[
a = \frac{\text{#negative samples correctly classified}}{\text{# total negative samples classified}} \times 100% .
\]

### Table 3: Experimental results of the testing geometric mean accuracy

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>LSGD</th>
<th>WLSGD</th>
<th>KSGD</th>
<th>WKSGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>78.86</td>
<td>72.19</td>
<td>87.60</td>
<td>94.19</td>
</tr>
<tr>
<td>Ijcnnl</td>
<td>51.42</td>
<td>78.35</td>
<td>55.14</td>
<td>72.36</td>
</tr>
<tr>
<td>Shuttle</td>
<td>0</td>
<td>40.51</td>
<td>5.15</td>
<td>94.54</td>
</tr>
<tr>
<td>Letter</td>
<td>23.14</td>
<td>67.51</td>
<td>75.63</td>
<td>86.04</td>
</tr>
<tr>
<td>Usps</td>
<td>90.83</td>
<td>91.35</td>
<td>94.66</td>
<td>94.03</td>
</tr>
<tr>
<td>Adult</td>
<td>0</td>
<td>76.08</td>
<td>9.16</td>
<td>73.21</td>
</tr>
</tbody>
</table>

Twenty trials were conducted for the four algorithms and the average results are shown in Table 3 and Table 4. Table 3 shows the performance comparison of accuracy of the four methods in the real-world problems; the testing geometric mean accuracy of WLSGD and WKSGD is higher than LSGD and KSGD methods in most datasets. Table 4 shows the performance comparison of average training and testing time of the four methods in the real-world problems. As observed from the Table 4, WLSGD and WKSGD methods compare to LSGD and KSGD methods with almost same learning speed in most datasets.
CONCLUSION

We focus on the large and imbalanced datasets effective classification problem, the stochastic gradient descent algorithms with SVM for imbalanced data classification are proposed. It consists of two stages. The first stage is to obtain the weight according to the size of positive and negative dataset. In the second stage, a fast learning algorithm on large datasets called the weighted stochastic gradient descent algorithm with SVM is proposed, which helps to reduce the hyperplane offset to the minority class, thus solve the large scale imbalanced data classification problems. Experiments on real datasets show that the proposed method is effective.

ACKNOWLEDGEMENT

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

REFERENCES


Available Online: http://saspjournals.com/sjpms