Voting-based SVM Ensemble with Map Reduce and Stochastic Gradient Descent

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

Abstract

Stochastic Gradient Descent (SGD) is an attractive choice for SVM training. In order to deal with the large-scale data linear classification problems, a method named Voting-based SVM Ensemble with MapReduce and Stochastic Gradient Descent (MR-SGD) is proposed. Firstly, to deal with the large-scale data classification problems, we use the MapReduce technique. Secondly, SVM optimization problem can be solved by stochastic gradient descent algorithm. Finally, the voting mechanism is used to ensemble several SVMs classifiers. Experimental results on datasets show that the proposed method is effective.

Keywords - *Stochastic gradient descent, Large-scale learning, Support vector machines, MapReduce, Voting Mechanism.*

I. INTRODUCTION

Recent advances in large-scale learning resulted in many algorithms for training SVMs using large data. Such as CVM [1], parallel SVMs and SGD. SGD is a simple and effective method, many works focus on designing variants of SGD that can reduce the variance and improve the complexity. Some popular methods include the Pegasos method [2], the stochastic gradient descent with Barzilai-Borwein update step for SVM [3], Budgeted Stochastic Gradient Descent for Large-Scale SVM Training [4], Bi-level stochastic gradient for large-scale support vector machine [5], and the stochastic variance reduced gradient method [6]. These methods are proven to converge linearly on strong convex problems. Pegasos performed stochastic gradient descent on the primal objective with a carefully chosen step size, which improves and guarantees convergence. Some recent works that discuss the improved approaches for SGD [7-12], 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.

A single classifier may not always provide a good classification performance. Ensembles of classifiers can overcome this limitation. An ensemble of classifiers is a set of multiple classifiers combining a number of weak learners to create a strong learner. There are many methods to study ensemble of classifiers for large-scale classification problem. [13] presented an ensemble of support vector machine for text-independent speaker recognition. Nasullah Khalid Alham et al [14] presented MRESVM that is a Map Reduce-based distributed SVM ensemble algorithm for scalable image annotation. The subsets to train a single SVM are selected using the method of bootstrapping. In addition, an ensemble algorithm has only one process of MapReduce. Sequential Minimal Optimization is used to training SVM. Ferhat et al [15] proposed a different MapReduce-based distributed SVM algorithm for binary classification.

In this paper, we focus on the large datasets linear classification problem, a method named voting-based SVM ensemble with MapReduce and SGD is proposed. The main contributions of this paper are as follows. Firstly, to deal with the large-scale data classification problems, we use the MapReduce technique. Secondly, SVM optimization problem can be solved by stochastic gradient descent algorithm. Finally, the voting mechanism is used to ensemble several SVMs classifiers. Experimental results show that the proposed method achieves faster convergence rate and higher classification accuracy in most cases of linear classification.

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

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

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 $\lambda \geq 0$ 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

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

where

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

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 $\mathbf{w}_{t+1}$.

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

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

The stochastic gradient descent for SVM is given in algorithm 1.

Algorithm 1 SGD for SVM

1. Input: data *S*, regularization parameter λ, a predetermined number *T* of iterations ;
2. Initialize: $\mathbf{w}_1 = \mathbf{0}$;
3. for $t = 1, \cdots, T$ do
4. choose $(\mathbf{x}_t, y_t)$ uniformly at random;
5. $\mathbf{w}_{t+1} \leftarrow (1-\eta_t\lambda)\mathbf{w}_t$
6. if $y_t \langle \mathbf{w}_t, \mathbf{x}_t \rangle < 1$ then
7. $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_{t+1} + \eta_t y_t \mathbf{x}_t$; //compute $\mathbf{w}_{t+1}$ according to the formulation (2)
8. else
9. $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_{t+1}$;
10. end if
11. end for
12. Output: $f_{t+1}(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}_{t+1}{}^T \mathbf{x})$.

III. VOTING-BASED SVM ENSEMBLE WITH MAPREDUCE AND STOCHASTIC GRADIENT DESCENT

Since the run-time of SGD algorithm does not depend directly on the size of the training set, it is especially suited for learning from large datasets.

We proposed an algorithm, which is voting-based SVM Ensemble with MapReduce and Stochastic Gradient Descent (MR-SGD). In MR-SGD, we utilize the Hadoop's Distributed File System and the MapReduce programming model to conquer the "big data" problem and to achieve a faster convergence rate than single SVM.

A. MapReduce Model

Here we introduce Apache Hadoop [16] that has two core components: Hadoop Distributed File System (HDFS), which based on [17] and MapReduce parallel computation programming model [18].

HDFS is used to store large data. It split data file into multiple chunks. Each chunk is stored in different data nodes.

MapReduce is a parallel processing programming model can handle the large data sets that are stored in HDFS. The basic function of the MapReduce model is iterate over the input, compute key/value pairs from each part of input, group all intermediate values by key, then iterate over the resulting groups and finally reduce each group. The model process in parallel. Map task is an initial transformation step, in which individual input records are processed in parallel. The system shuffles and sorts the map outputs and transfers them to the reduce tasks. Reduce task is a summarization step, in which all associated records are processed together by a single entity.

B. MR-SGD

There are two properties about SGD algorithm:

1. The run-time scales linearly with the number of iterations and does not depend on the number of the training size.

2. It proceeds by iteratively choosing a labeled example randomly from training set. If the number of iteration is less than the training set size, all of the instances that are chosen to train classifier are just a partition of the training set.

Given the properties of SGD described above, we can utilize the HDFS to store training set and use MapReduce programming model to select multiple subset of the training set and to train SVM on each subset in parallel using a cluster of computers. Finally, we aggregate all the trained SVMs using Voting Mechanism to predict a testing instance.

The implementation details of MR-SGD describes as follows.

1) First, we pick up *k* subsets from training set where *k* should be odd to support the voting Mechanism. The size of subset that required training an effective SVM ensemble is different with different training set.

Assume there are *m* training examples and want to randomly choose *d* examples as a subset. Map task deal with each instance in the training set. It loops *k* times to decide whether this instance should be chosen into the corresponding subset. In the *i*th ($i \in \{1,\ldots, k\}$) loop, it will generate a random integer from 1 to *m*. If the random integer is less than *d*, then the instance will be put into the *i*th subset to train an SVM in Reduce task.

2) Then Reduce tasks use the subsets that are chosen in Map tasks to train SVMs with SGD algorithm. There is *k* SVMs just as we set in the first step.

3) Finally, we aggregate *k* SVMs using Voting Mechanism to decide the label of the test samples.

IV. EXPERIMENTAL RESULTS

In this section, we perform some experiments that demonstrate the efficacy of our algorithm. The basic SGD algorithm is Pegasos [2]. To evaluate the classification accuracy and convergence rate of Pegasos and MR-SGD, several datasets are used to illustrate in the linear kernel situations. MR-SGD experiments are carried out on a cluster contains six machines. Each of machines has four E5-2609 2.50GHz processors and 4GB RAM memory. The operating system is the CentOS-6.4. Apache Hadoop is the 2.4.1 version.

We tested the performance of MR-SGD and Pegasos on three large datasets and four standard real datasets, three large datasets are derived from Pascal Large Scale Learning Challenge [19], four standard real datasets are downloaded from LIBSVM website [20]. For the MR-SGD, we set the number of training examples in each subset to 3,000. 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 and the parameters are given in Table 1.

Table 1: Datasets and Parameters

Dataset	#Training	#Testing	#Features	λ
Alpha	400,000	100,000	500	4.0E-7
Gamma	400,000	100,000	500	1.0E-6
Delta	400,000	100,000	500	8.0E-8
Usps	7,291	2,007	256	1.35E-5
Mnist	60,000	10,000	780	1.67E-5
Letter	15,000	5,000	16	1.55E-6
Adult	32,561	16,281	123	3.07E-5

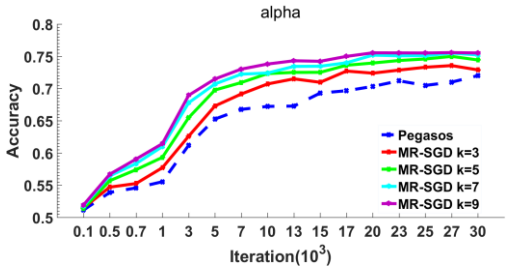

Fig. 1 Testing accuracy on Alpha dataset with linear kernel

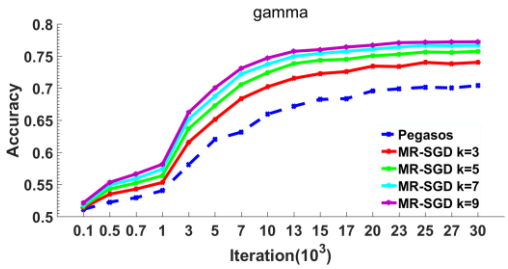

Fig. 2 Testing accuracy on Gamma dataset with linear kernel

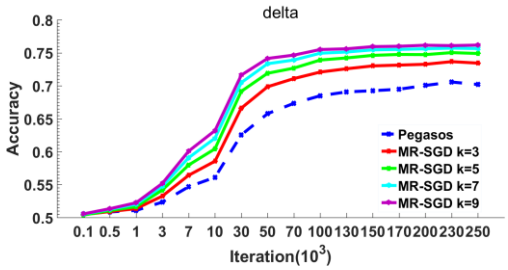

Fig. 3 Testing accuracy on Delta dataset with linear kernel

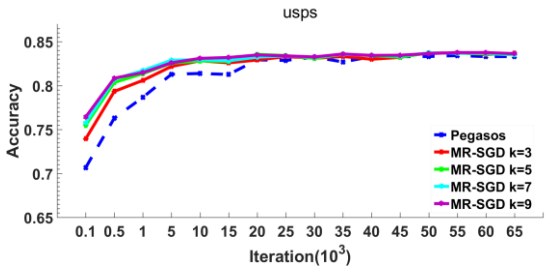

Fig. 4 Testing accuracy on Usps dataset with linear kernel

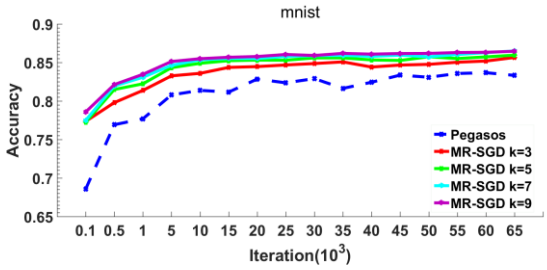

Fig. 5 Testing accuracy on Mnist dataset with linear kernel

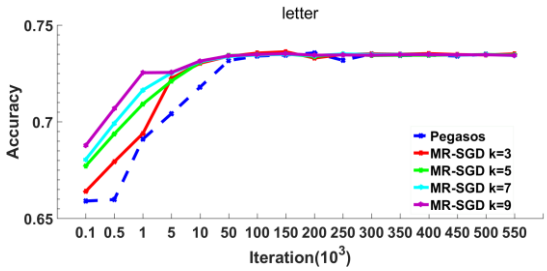

Fig. 6 Testing accuracy on Letter dataset with linear kernel

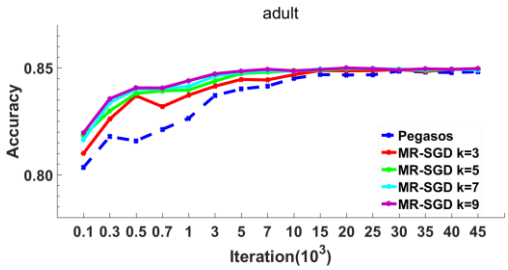

Fig. 7 Testing accuracy on Adult dataset with linear kernel

Figures 1-7 shows the convergence rate both Pegasos and MR-SGD with the number of iteration growing. Figures 1-7 shows that MR-SGD for linear kernel has a faster convergence rate than Pegasos on

seven datasets. The classification accuracy of MR-SGD is slightly higher than Pegasos on seven datasets.

V. CONCLUSION

We focus on the large datasets effective linear classification problem, a method named voting-based SVM ensemble with MapReduce and SGD is proposed. The main contributions of this paper are as follows. Firstly, to deal with the large-scale data classification problems, we use the MapReduce technique. Secondly, SVM optimization problem can be solved by stochastic gradient descent algorithm. Finally, the voting mechanism is used to ensemble several SVMs classifiers. Experimental results show that the proposed method achieves faster convergence rate and higher classification accuracy in most cases of linear classification. Future work will be study large-scale non-linear kernel SVM ensemble with MapReduce and SGD.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] W. Tsang, J. T. Kwok, and P. M. Cheung, Core vector machines, fast SVM training on very large data sets. Journal of Machine Learning Research. 6, 2005, 363-392.
[2] Shalev-Shwartz, Y. Singer, N. Srebro, et al, Pegasos: Primal Estimated sub-Gradient Solver for SVM, Mathematical Programming, 127(1), 2011, 3-30.
[3] Krzysztof Sopyla, Pawel Drozda, Stochastic Gradient Descent with Barzilai-Borwein update step for SVM, Information Sciences, 316, 2015, 218-233.
[4] 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.
[5] Nicolas Couellan, Wenjuan Wang, Bi-level stochastic gradient for large scale support vector machine, Neurocomputing, 153, 2015, 300-308.
[6] R. Johnson and T. Zhang, Accelerating Stochastic Gradient Descent using predictive variance reduction. In Advances in Neural Information Processing Systems, 2013, 315-323.
[7] A. Bordes, L. Bottou, P. Gallinari, SGD-QN: careful quasi-Newton stochastic gradient descent, J. Mach. Learn, 10, 2009, 1737-1754.
[8] A. Bordes, L. Bottou, P. Gallinari, et al, Sgdqn is less careful than expected, J. Mach. Learn, 11, 2010, 2229-2240.
[9] Shai Shalev-Shwartz, Tong Zhang, Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization, Math. Program., 155, 2016, 105-145.
[10] Shalev-Shwartz, Zhang, et al, Stochastic dual coordinate ascent methods for regularized losss minimization, J. Mach. Learn, 14, 2013, 567-599.
[11] 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.
[12] 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.
[13] 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.
[14] Nasullah Khalid Alham, Maozhen Li, Yang Liu, Man Qi, A MapReduce-based distributed SVM ensemble for scalable image classification and annotation. Computers and Mathematics with Applications, 66, 2013, 1920-1934.
[15] Ferhat Ozgur CATAK, Mehmet Erdal BALABAN, A MapReduce-based distributed SVM algorithm for binary classification, Turkish Journal of Electrical & Computer Science, 2013, 863-873
[16] Apache Hadoop, http://hadoop.apache.org/
[17] K Shvachko, H Kuang, S Radia, R Chansler, The Hadoop Distributed File System, IEEE Symposium on Mass Storage System & Technologies, 11, 2010, 1-10.
[18] J. Dean and S. Ghemawat, Mapreduce: simplified data processing on large clusters, Communications of the ACM, 51(1), 2008, 107–113.
[19] S. Sonnenburg, V. Franc, E.Y. Tov, M. Sebag, PASCAL large-scale learning challenge, 2008.
[20] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2016.