Parallel Bag-SVM-SGD for classifying very high-dimensional and large-scale multi-class datasets

Thanh-Nghi Do
College of Information Technology
Can Tho University
UMI UMMISCO 209 (IRD/UPMC)
dtnghi@cit.ctu.edu.vn

Nguyen-Khang Pham
College of Information Technology
Can Tho University
pnkhang@cit.ctu.edu.vn

The-Phi Pham
College of Information Technology
Can Tho University
ptphi@cit.ctu.edu.vn

Minh-Thu Tran-Nguyen
College of Information Technology
Can Tho University
tnmthu@cit.ctu.edu.vn

Huu-Hoa Nguyen
College of Information Technology
Can Tho University
nhhoa@ctu.edu.vn

ABSTRACT

We propose the parallel bagging support vector machines using stochastic gradient descent (Bag-SVM-SGD) on multi-core computers for effectively classifying very-high-dimensional and large-scale multi-class datasets. The Bag-SVM-SGD learns in a parallel way from under-sampling training dataset to create ensemble binary SVM-SGD classifiers used in the One-Versus-All (OVA) multi-class strategy for performing text/image classfication tasks with million of datapoints in millions of dimensions and thousands of classes. The numerical test results on four large scale multi-class datasets (ImageNet, LSHTC4, Book) show that our Bag-SVM-SGD algorithm is faster and more accurate than the state-of-the-art linear algorithm LIBLINEAR. An example of its effectiveness is given with an accuracy of 62.41% obtained in the classification of LSHTC4 dataset having 728,067 datapoints in 1,617,900 dimensions into 2,713 classes in 104.15 minutes using a PC Intel(R) Core i7-4790 CPU, 3.6 GHz, 4 cores.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Support vector machines**;

KEYWORDS

Support Vector Machines (SVM), Stochastic gradient descent (SGD), Bagging method, High-dimensional and large-scale multi-class data classification

ACM Reference Format:
Thanh-Nghi Do, Nguyen-Khang Pham, The-Phi Pham, Minh-Thu Tran-Nguyen, and Huu-Hoa Nguyen. 2017. Parallel Bag-SVM-SGD for classifying very high-dimensional and large-scale multi-class datasets. In *Proceedings of ACM SIGGRAPH ASIA Workshop, Thailand, Nov. 2017 (D2AT),* 8 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
D2AT, Nov. 2017, Thailand
© 2017 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . . $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

There are more and more multimedia data stored electronically, with increasing number of internet users and mobile internet access sharing videos, songs or photos. There are more than one billion daily active users - nearly one-third of all people on the Internet (around 46% of the world population) - on Youtube and Facebook (Twitter, Amazon and Yahoo! have even more). This leads to very huge amount of data, there is a need for high performance classification algorithms in order to help us find what we are looking for. The automatic classification of texts/images plays an important role in information management tasks, including auto-tagging emails, news, images, detecting or discovering topics, filtering. The popular frameworks for text/image classification [Fabrizio 2002; Manning et al. 2008; Sivic and Zisserman 2003] involves the main steps as follows: the feature extraction of texts/images, encoding features, the representation of texts/images with the Bag-of-Words (BoW [Salton et al. 1975]) model and training supervised classifiers such as naïve Bayes, decision trees, support vector machines (SVM [Vapnik 2000]). The dictionary has millions vocabulary words. Therefore, the BoW representation brings-out datasets with a very large number of dimensions. And then, the SVM model is suited for classifying this kind of data without any feature selection or reduction methods [Do 2014; Do and Tran-Nguyen 2016; Doan et al. 2015; Dumais et al. 1998; Joachims 1998; Lewis and Gale 1994; Li and Perona 2005; Sivic and Zisserman 2003; Wu 2012].

Nevertheless, the emergence of new benchmarks of text/image classification tasks yields huge classification challenges of very-high-dimensional and large-scale multi-class. For example, ImageNet dataset [Deng et al. 2010, 2009] has more than 14 million images and 21,841 classes. LSHTC4 dataset [Partalas et al. 2015] has 2.4 million documents and 325,000 classes.

Our Book dataset consists of 115,000 abtracts and 661 subjects. It is very hard for any machine learning algorithm to be able to handle such datasets.

This challenge motivates us to study the bagging support vector machines using stochastic gradient descent (Bag-SVM-SGD) for effectively dealing with very-high-dimensional and large-scale multi-class datasets. We propose to extend the binary SVM-SGD algorithm [Bottou and Bousquet 2008; Shalev-Shwartz et al. 2007] to develop the bagging binary SVM-SGD classifiers learnt from under-sampling training dataset. These Bag-SVM-SGD models trained in the parallel way on multi-core computers are used in the One-Versus-All multi-class strategy to perform text/image classfication tasks with million of datapoints in millions of dimensions and thousands of classes. The numerical test results on large scale multi-class datasets (ImageNet, LSHTC4, Book) show that our Bag-SVM-SGD algorithm is faster and more accurate than the state-of-the-art linear algorithm, LIBLINEAR [Fan et al. 2008].

The paper is organized as follows. Section 2 briefly introduces the SVM-SGD algorithm and our proposed Bag-SVM-SGD algorithm for dealing with very high-dimensional and large-scale multi-class datasets in multi-core computers. Section 3 shows the experimental results. We then conclude in section 4.

2 BAGGING SUPPORT VECTOR MACHINES USING THE STOCHASTIC GRADIENT DESCENT FOR LARGE-SCALE MULTI-CLASS DATSETS

2.1 Support vector machine for binary classification

Let us consider a binary classification problem with the dataset $D = [X, Y]$ consisting of m datapoints $X = \{x_1, x_2, \ldots, x_m\}$ in the n-dimensional input space R^n, having corresponding labels $Y = \{y_1, y_2, \ldots, y_m\}$ being ± 1. The SVM algorithm porposed by Vapnik [Vapnik 2000] tries to find the best separating plane (denoted by the normal vector $w \in R^n$), i.e. furthest from both class $+1$ and class -1. It is accomplished through the maximization of the margin (or the distance) between the supporting planes for each class. The margin between these supporting planes is $2/\|w\|$ (where $\|w\|$ is the 2-norm of the vector w). Any point x_i falling on the wrong side of its supporting plane is considered to be an error, its error distance denoted by $z_i = 1 - y_i(w.x_i) \geq 0$. The error z_i is rewritten by $L(w, [x_i, y_i]) = max\{0, 1 - y_i(w.x_i)\}$. And then, SVM has to simultaneously maximize the margin and minimize the error. The SVM pursues these goals with the unconstrained problem (1).

$$\min \ \Psi(w, [X, Y]) = \frac{\lambda}{2}\|w\|^2 + \frac{1}{m}\sum_{i=1}^{m} L(w, [x_i, y_i]) \qquad (1)$$

where λ is a positive constant used to tune the trade-off between the margin size and the error.

And then, [Bottou and Bousquet 2008; Shalev-Shwartz et al. 2007] proposed the stochastic gradient descent (SGD) method to solve the unconstrained problem (1). The SGD for SVM (denoted by SVM-SGD) updates w on T epochs with a learning rate η. For each epoch t, the SVM-SGD uses a single randomly received datapoint (x_i, y_i) to compute the sub-gradient $\nabla_t \Psi(w, [x_i, y_i])$ and update w_{t+1} as follows:

$$\begin{aligned} w_{t+1} &= w_t - \eta_t \nabla_t \Psi(w, [x_t, y_t]) \\ &= w_t - \eta_t(\lambda w_t + \nabla_t L(w, [x_t, y_t])) \end{aligned} \qquad (2)$$

$$\begin{aligned} \nabla_t L(w, [x_t, y_t]) &= \nabla_t max\{0, 1 - y_t(w.x_t)\} \\ &= \begin{cases} -y_t x_t & if\ y_t(w.x_t) < 1 \\ 0 & otherwise \end{cases} \end{aligned} \qquad (3)$$

The SVM-SGD using the update rule (2) is described in algorithm 1.

Algorithm 1: SVM-SGD(D, λ, T) for binary classification

input :
training dataset D
positive constant $\lambda > 0$
number of epochs T
output:
hyperplane w

```
begin
    init w_1 s.t. ||w_1|| ≤ 1/√λ
    for t ← 1 to T do
        randomly pick a datapoint [x_t, y_t] from
        training dataset D
        set η_t = 1/(λt)
        if (y_i(w_t.x_i) < 1) then
            w_{t+1} = w_t − η_t(λw_t − y_i x_i)
        else
            w_{t+1} = w_t − η_t λ w_t
        end
    end
    return w_{t+1}
end
```

As mentioned in [Bottou and Bousquet 2008; Shalev-Shwartz et al. 2007], the SVM-SGD algorithm quickly converges to the optimal solution due to the fact that the unconstrained problem (1) is convex optimization problems on very large datasets. The algorithmic complexity of SVM-SGD is linear with the number of datapoints. An example of its effectiveness is given with the binary classification of 780,000 datapoints in 47,000-dimensional input space in 2 seconds on a PC and the test accuracy is similar to standard SVM, e.g. LIBLINEAR [Fan et al. 2008].

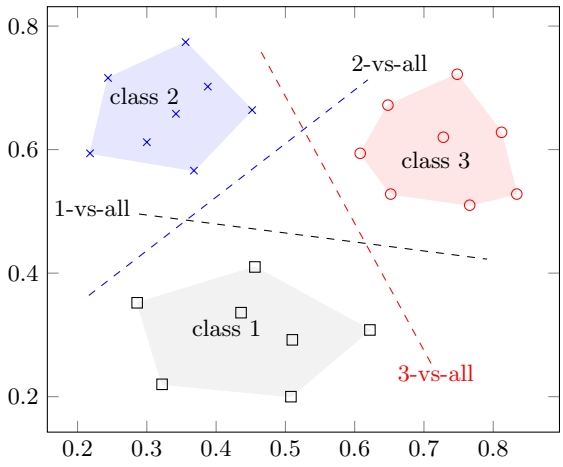

Figure 1: Multi-class SVM (One-Versus-All)

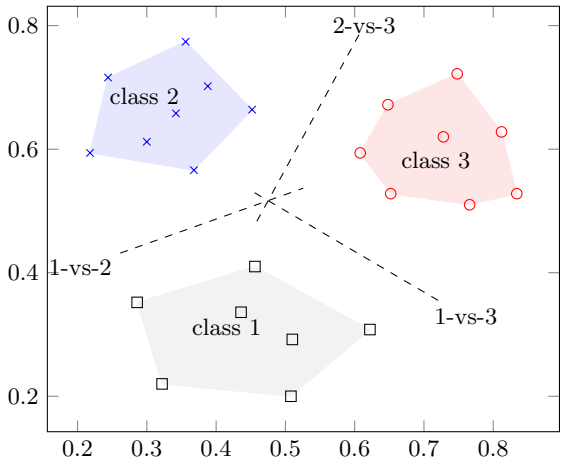

Figure 2: Multi-class SVM (One-Versus-One)

2.2 Support vector machines for multi-class

There are two strategies to extend the binary SVM solver for dealing with the multi-class problems (c classes, $c \geq 3$). The first one is considering the multi-class case in one optimization problem [Guermeur 2007; Weston and Watkins 1999]. The second one is decomposing multi-class into a series of binary SVMs, including One-Versus-All [Vapnik 2000], One-Versus-One [Kreßel 1999]. In practice, the most popular methods are One-Versus-All (ref. LIBLINEAR [Fan et al. 2008]), One-Versus-One (ref. LibSVM [Chang and Lin 2011]) and are due to their simplicity. The One-Versus-All strategy (as illustrated in Figure 1) builds c different binary SVM models where the i^{th} one separates the i^{th} class from the rest. The One-Versus-One strategy (as illustrated in Figure 2) constructs $c(c-1)/2$ binary SVM models for all the binary pairwise combinations of the c classes. The class is then predicted with the largest distance vote.

2.3 Parallel bagging SVM-SGD for large-scale multi-class

When dealing with very large number of classes, e.g. $c = 1,000$ classes, the One-Versus-One strategy is too expensive because it needs training $499,500$ of binary classifiers and using them in the classification (compared to $1,000$ binary models learned by the One-Versus-All strategy). Therefore, the One-Versus-All strategy is suited for handling this case. And then, we propose to use the One-Versus-All approach to train c binary SVM-SGD classifiers. However, the multi-class SVM-SGD algorithm using One-Versus-All leads to the two problems:

(1) the SVM-SGD algorithm deals with the imbalanced datasets for building binary classifiers,
(2) the SVM-SGD algorithm also takes very long time to train very large number of binary classifiers in sequential mode using a single processor,

Due to these problems, we propose the parallel bagging SVM-SGD algorithm (denoted by Bag-SVM-SGD) being able to efficiently handle the large number of datapoints in very-high dimensional input space and large-scale multi-class on standard personal computers (PCs). The first one is to build ensemble binary classifiers with under-sampling strategy. The second one is to parallelize the training task of all binary classifiers with multi-core machines.

Bagging binary SVM-SGD classifier

In the multi-class SVM-SGD algorithm using One-Versus-All approach, the learning task of binary SVM-SGD classifier is try to separate the c_i class (positive class) from the $c-1$ other classes (negative class). For very large number of classes, this leads to the extreme unbalance between the positive and the negative class. The problem of binary SVM-SGD comes from **line 4** of algorithm 1. Given a classification problem with $1,000$ classes, the probability for a positive datapoint sampled is very small (about 0.001) compared with the large chance for a negative datapoint sampled (e.g. 0.999). And then, the binary SVM-SGD classifier focuses mostly on the negative datapoints. Therefore, the binary SVM-SGD classifier has difficulty to separate the positive class from the negative class, well-known as the class imbalance problems.

One of the most popular solutions for dealing with the imbalanced data [Japkowicz 2000; Visa and Ralescu 2005; Weiss and Provost 2003] is to change the data distribution, including over-sampling the minority class [Chawla et al. 2003] or under-sampling the majority class [Liu et al. 2009; Ricamato et al. 2008]. Nevertheless, over-sampling the minority class is very expensive due to large datasets with millions datapoints.

Given the training dataset D consists of the positive class D_+ ($|D_+|$ is the cardinality of the positive class c_i) and the negative class D_- ($|D_-|$ is the cardinality of the negative class). Our bagging binary SVM-SGD trains κ SVM-SGD classifiers $\{w_1, w_2, \ldots, w_\kappa\}$ from mini-batch using under-sampling the majority class (negative class) to separate the positive class c_i from the negative class. Since the original bagging [Breiman 1996] uses bootstrap sampling from the

training dataset without regard to the class distribution. Our bagging SVM-SGD follows the idea of bagging in more appropriate strategy for dealing with class imbalanced. At the i^{th} iteration, the mini-batch mB_i includes n_p datapoints randomly sampling without replacement from the positive class D_+ and $n_p\sqrt{\frac{|D_-|}{|D_+|}}$ datapoints sampling without replacement from the negative class D_-, and then the learning algorithm 1 learns w_i from mB_i. Such a mini-batch improves the chance for a positive datapoint sampled in learning algorithm 1. The Bag-SVM-SGD averages all classifiers $\{w_1, w_2, \ldots, w_\kappa\}$ to create the final model w for separating the class c_i from other ones. The Bag-SVM-SGD is described in algorithm 2.

Algorithm 2: Bag-SVM-SGD$(c_i, D, \lambda, T, \kappa)$ classifier used in the One-Versus-All approach of large-scale multi-class SVM

input :
positive class c_i versus other classes
training dataset D
positive constant $\lambda > 0$
number of epochs T
number of SVM-SGD models κ

output:
hyperplane w

begin
  splitting training dataset D into
  the positive data D_+ (class c_i) and the negative data D_-
  for $i \leftarrow 1$ **to** κ **do**
    creating a mini-batch mB_i by sampling without replacement n_p datapoints from D_+ and D'_- from D_- (with $|D'_-| = n_p\sqrt{\frac{|D_-|}{|D_+|}}$)
    $w_i =$ **SVM-SGD**(mB_i, λ, T)
  end
  return $w = \frac{1}{\kappa}\sum_{i=1}^{\kappa} w_i$
end

Parallel training of Bag-SVM-SGD

The Bag-SVM-SGD algorithm independently trains c binary classifiers for c classes in multi-class SVM-SGD. This is a nice property for parallel learning. The main idea is to learn c binary classifiers in parallel to speedup training tasks for large-scale multi-class datasets. The simplest development of parallel Bag-SVM-SGD described in algorithm 3 is based on the shared memory multiprocessing programming model OpenMP on multi-core computers.

Algorithm 3: Parallel training ensemble binary Bag-SVM-SGD classifiers in the One-Versus-All approach of large-scale multi-class SVM

input :
training dataset D with c classes
positive constant $\lambda > 0$
number of epochs T
number of SVM-SGD models κ

output:
hyperplanes $\{w_1, w_2, \ldots, w_c\}$

begin
  #pragma omp parallel for schedule(dynamic)
  for $c_i \leftarrow 1$ **to** c **do** `/* class` c_i `*/`
    $w_{c_i} =$ **Bag-SVM-SGD**$(c_i, D, \lambda, T, \kappa)$
  end
end

3 EVALUATION

In order to evaluate the performance (accuracy and training time) of the Bag-SVM-SGD algorithm for classifying a large amount of data in very-high-dimensional and large-scale multi-class, we have implemented the Bag-SVM-SGD in C/C++, OpenMP [OpenMP Architecture Review Board 2008]. We are interested in the best state-of-the-art linear SVM algorithm, LIBLINEAR (a library for large linear classification [Fan et al. 2008], the parallel version on multi-core computers). Therefore, we here report the comparison of the classification performance obtained by the Bag-SVM-SGD and the LIBLINEAR.

All experiments are run on a PC with Linux Fedora 20, Intel(R) Core i7-4790 CPU, 3.6 GHz, 4 cores and 32 GB main memory.

3.1 Description of datasets

The Bag-SVM-SGD algorithm is designed for dealing with a large amount number of data in very-high-dimensional and large-scale multi-class, so we have evaluated its performance on the four following datasets.

ImageNet 100

This dataset consists of the 100 largest classes from ImageNet [Deng et al. 2010, 2009], including 183,116 images. In each class, we sample 50% images for training and 50% images for testing (with random guess 1%). First, we construct BoW of every image using dense SIFT descriptor (extracting SIFT on a dense grid of locations at a fixed scale and orientation [Lowe 2004; Vedaldi and Fulkerson 2010]) and 5,000 codewords. Then, we use feature mapping from [Vedaldi and Zisserman 2012] to get the high-dimensional image representation in 15,000 dimensions. This feature mapping has been proven to give a good image classification performance with linear classifiers [Vedaldi and Zisserman 2012].

ILSVRC 2010

This dataset contains 1,000 classes from ImageNet [Deng et al. 2010, 2009], including 1.2 million images for training, 50 thousand images for validation and 150 thousand images for testing (with random guess 0.1%). We use BoW feature set provided by [Deng et al. 2010] and the method reported in [Wu 2012] to encode every image as a vector in 21,000 dimensions. We take roughly 900 images per class for training dataset, so the total training images is 887,816.

Book collection

Book dataset is the real book collection at the Learning Resource Center of Can Tho University in Vietnam. It consists of 114998 books in a quadruplet format:

$< Title; Abstract; Keywords; Subject >$.

The aim is to automatically assign the subject to the book based on the information $< Title; Abstract; Keywords >$. Due to the book information in Vietnamese and in English, there are not only one-syllable words but also multiple syllables ones. We use JvnTextPro [Nguyen et al. 2010] well-known as a good Vietnamese word segmentation, to perform the word splitting. The dictionary has 89,821 vocabulary words. The representation of books in the BoW model brings out the table with 114,998 rows, 89,821 columns. Furthermore, there are 661 subjects. Therefore, it yields huge classification challenges of very-high-dimensional and large-scale multi-class dataset. The dataset is randomly divided into training set with 100,000 rows and testing set with 14,998 rows (with random guess $\sim 0.1513\%$).

LSHTC4 dataset

The LSHTC4 dataset [Partalas et al. 2015] is a benchmark for large-scale text classification. The dataset originates from the DBpedia site. The LSHTC4 is represented in the BoW model. The dictionary has 1,617,900 vocabulary words. The categories in the DBpedia site have the complex relationships. Since the LSHTC4 is multi-label benchmark, a text (datapoint) may belong to more than one category with respect to the hierarchy of categories. We convert the multi-label LSHTC4 to the multi-class one. The top category is assigned for a text. Furthermore, the dataset only includes the categories having more than 100 individuals. We obtain the new multi-class LSHTC4 dataset with 728,067 datapoints in 1,617,900 dimensions into 2,713 classes. And then, the dataset is randomly divided into training set with 628,067 rows and testing set with 100,000 rows (with random guess $\sim 0.0369\%$).

3.2 Tuning parameters

With such large datasets in very high-dimensional, linear SVM models have given competitive performances compared to non-linear classifiers but training and testing are much faster [Doan et al. 2013; Yuan et al. 2012]. That is why we propose to use linear SVM models for classifying these large datasets in the experiments.

The training set is used to build the classification model and tune the parameters. Then, the classification results are reported on the testing set using the resulting models.

The positive constant $C = 100,000$ (a trade-off between the margin size and the errors in learning SVM algorithms, the same tuning in [Do 2014; Do and Tran-Nguyen 2016; Doan et al. 2015]) was used in LIBLINEAR.

Our Bag-SVM-SGD algorithm learns $\kappa = 50$ binary SVM-SGD classifiers and regularization term $\lambda = 0.00002$ to separate one class from other ones.

Due to the PC (Intel(R) Core i7-4790 CPU, 4 cores) used in the experimental setup, we try to vary the number of OpenMP threads (1, 4 threads) for all training tasks.

Table 1: Training time (minutes) on ImageNet 100

	# OpenMP threads	
Algorithm	1	4
LIBLINEAR	63.42	30.49
Bag-SVM-SGD	26.13	7.00

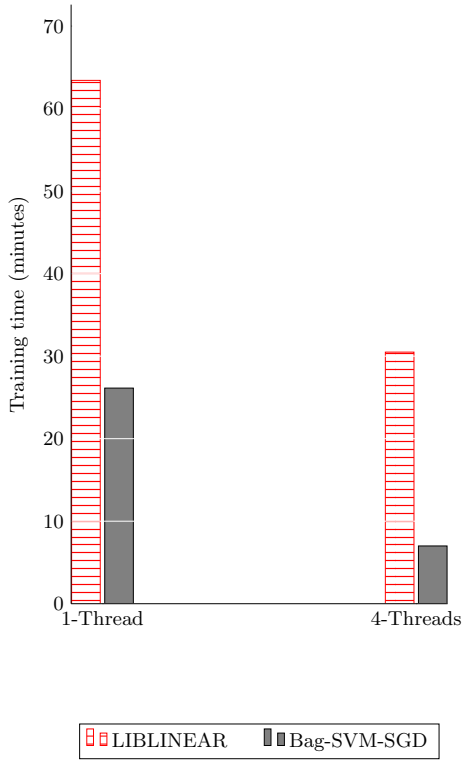

Figure 3: Training time (minutes) on ImageNet 100

3.3 Classification results

Firstly, we would like to compare the training time of the Bag-SVM-SGD to the LIBLINEAR on four large-scale multi-class datasets as described above.

Training time

Table 1 and Fig. 3 present the training time of algorithms for ImageNet 100. It shows that the Bag-SVM-SGD is 4.36 times faster than LIBLINEAR with 4 OpenMP threads.

For ILSVRC 2010 having large amount of images (1 million images) and very large number of classes (1,000 classes), the LIBLINEAR using 4 OpenMP threads takes 1,037.00 minutes to train the classification model for this dataset. Our Bag-SVM-SGD algorithm performs the learning task in 37.04 minutes with the same setting (4 OpenMP threads). This indicates that the Bag-SVM-SGD is 28 times faster than the LIBLINEAR.

For Book collection having medium number of texts (100,000 texts) and large-scale multi-class (661 classes), the training

Table 2: Training time (minutes) on ILSVRC 2010

	# OpenMP threads	
Algorithm	1	4
LIBLINEAR	3106.48	1037.00
Bag-SVM-SGD	145.14	37.04

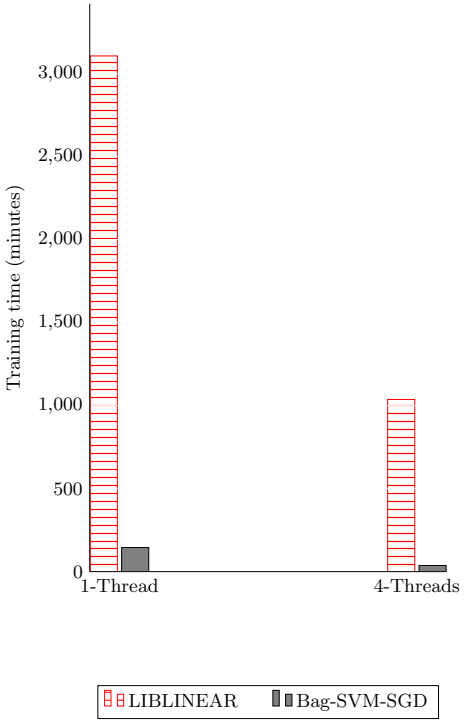

Figure 4: Training time (minutes) on ILSVRC 2010

Table 3: Training time (minutes) on Book collection

	# OpenMP threads	
Algorithm	1	4
LIBLINEAR	12.34	5.57
Bag-SVM-SGD	1.47	0.46

time of the LIBLINEAR using 4 OpenMP threads is 5.57 minutes, compared to 0.46 minutes of the Bag-SVM-SGD. Once again, the Bag-SVM-SGD is 12.11 times faster than the LIBLINEAR.

LSHTC4 is the bigest dataset with 600,000 texts in 1,617,900 dimensions into 2,713 classes. The LIBLINEAR can not deal with LSHTC4 due to the Segmentation fault (core dumped) error. The Bag-SVM-SGD has finished the training task in 381.59 minutes (using 1 OpenMP thread) and 104.15 minutes (using 4 OpenMP threads).

Training time reported in tables 1, 2, 3 and 4 show that our Bag-SVM-SGD can reduce learning time linearly with the number of cores used in training tasks.

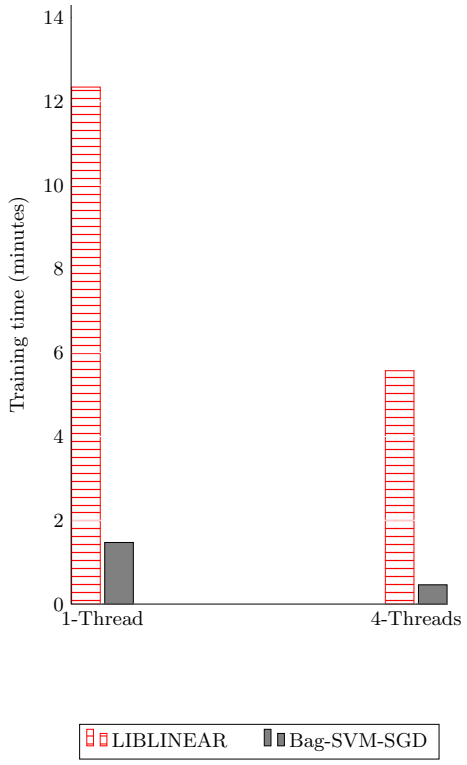

Figure 5: Training time (minutes) on Book collection

Table 4: Training time (minutes) on LSHTC4

	# OpenMP threads	
Algorithm	1	4
LIBLINEAR	N/A	N/A
Bag-SVM-SGD	381.59	104.15

Classification accuracy

Table 5 and Fig. 7 present the classification results in terms of accuracy. The Bag-SVM-SGD achieves very competitive correctness compared to the LIBLINEAR on ImageNet 100 and Book collection datasets.

ILSVRC 2010 is a very large-scale multi-class dataset. Thus, it is very difficult for many state-of-the-art algorithms to obtain a high rate in classification performance. In particular, with the feature set provided by ILSVRC 2010 competition the state-of-the-art system [Deng et al. 2010] reports an accuracy of approximately 19% (it is far above random guess, 0.1%). More recent, Balanced trees [Mai et al. 2017] can achieve 20.45% correctness. Our Bag-SVM-SGD algorithm gives a higher accuracy rate than [Deng et al. 2010] and with the same feature set (22.61% versus 19%). The improvement is about 3.61%. The Bag-SVM-SGD also outperforms the LIBLINEAR with an improvement of 2.5% correctness rate.

Only Bag-SVM-SGD has finished the training task on LSHTC4 dataset with an accuracy of 62.28%.

Table 5: Overall classification accuracy (%)

	Dataset			
Algorithm	ImageNet 100	ILSVRC 2010	Book	LSHTC4
LIBLINEAR	54.07	21.11	64.61	N/A
Bag-SVM-SGD	53.70	22.61	64.69	62.41

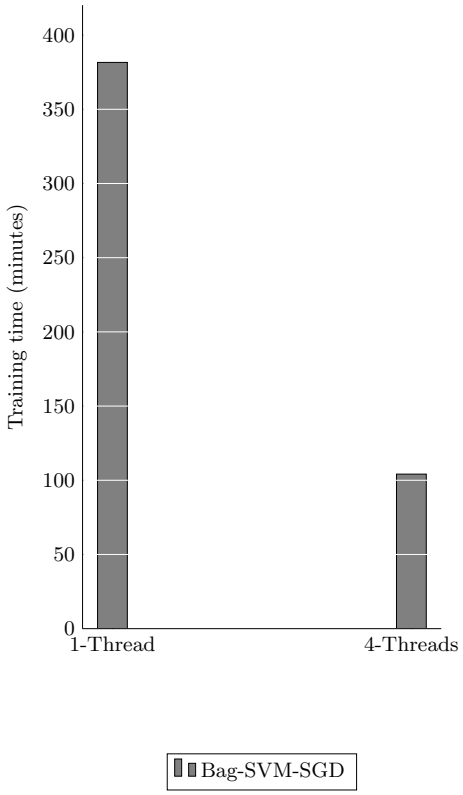

Figure 6: Training time (minutes) on LSHTC4

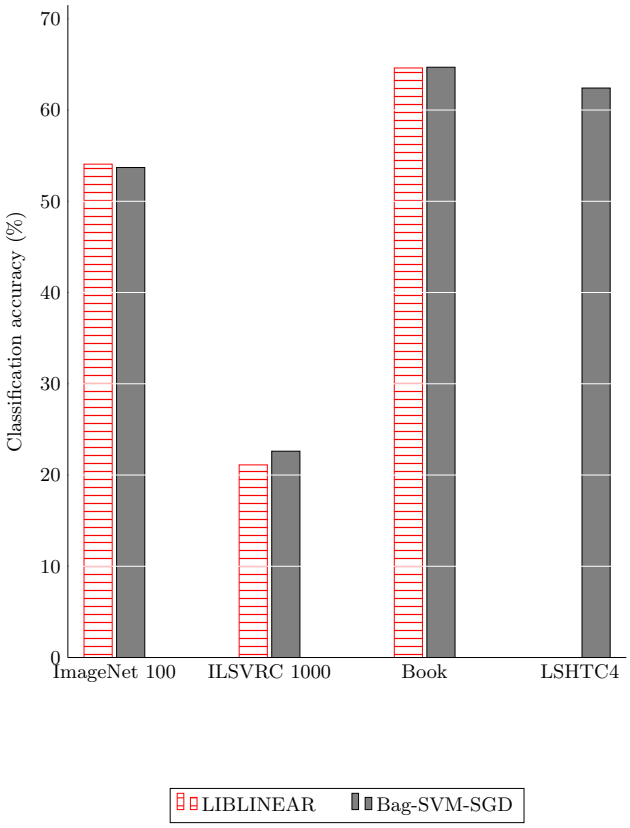

Figure 7: Overall classification accuracy (%)

These results show that our Bag-SVM-SGD has a great ability to scale-up to a large amount of datapoints in very-high-dimensional input space and large-scale multi-class.

4 CONCLUSION AND FUTURE WORKS

We have presented the parallel bagging support vector machines using stochastic gradient descent (Bag-SVM-SGD) on multi-core computers that achieves high performances for dealing with a large amount of data in very-high-dimensional and large-scale multi-class. The main idea of the Bag-SVM-SGD is to learn in a parallel way from under-sampling training dataset to create ensemble binary SVM-SGD classifiers used in the One-Versus-All (OVA) multi-class strategy for performing classfication tasks with million of datapoints in millions of dimensions and thousands of classes. The numerical test results on four large scale multi-class datasets (ImageNet, LSHTC4, Book) show that our Bag-SVM-SGD algorithm is faster and more accurate than the state-of-the-art linear algorithm LIBLINEAR.

In the near future, we intend to develop a distributed implementation for large scale processing on an in-memory cluster-computing platform, Apache Spark [Zaharia et al. 2010] (running times or up to 100x faster than Hadoop MapReduce, or 10x faster on disk).

ACKNOWLEDGMENTS

This work has been funded by the AniAge project (High Dimensional Heterogeneous Data based Animation Techniques for Intangible Cultural Heritage Southeast Asian Digital Content). The authors would like to particularly thank for the AniAge's support.

REFERENCES

L. Bottou and O. Bousquet. 2008. The Tradeoffs of Large Scale Learning. In *Advances in Neural Information Processing Systems*, J.C. Platt, D. Koller, Y. Singer, and S. Roweis (Eds.), Vol. 20. NIPS Foundation (http://books.nips.cc), 161–168.

L. Breiman. 1996. Bagging Predictors. *Machine Learning* 24, 2 (1996), 123–140.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 27 (2011), 1–27.

Nitesh V. Chawla, Ar Lazarevic, Lawrence O. Hall, and Kevin W. Bowyer. 2003. SMOTEBoost: improving prediction of the minority class in boosting. In *In Proceedings of the Principles of Knowledge Discovery in Databases, PKDD-2003*. 107–119.

Jia Deng, Alexander C. Berg, Kai Li, and Fei-Fei Li. 2010. What Does Classifying More Than 10, 000 Image Categories Tell Us?. In *Computer Vision - ECCV 2010 - 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V*. 71–84.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. 248–255.

Thanh-Nghi Do. 2014. Parallel multiclass stochastic gradient descent algorithms for classifying million images with very-high-dimensional signatures into thousands classes. *Vietnam J. Computer Science* 1, 2 (2014), 107–115.

Thanh-Nghi Do and Minh-Thu Tran-Nguyen. 2016. Incremental Parallel Support Vector Machines for Classifying Large-Scale Multi-class Image Datasets. In *Future Data and Security Engineering - Third International Conference, FDSE 2016, Can Tho City, Vietnam, November 23-25, 2016, Proceedings*. Springer, 20–39.

Thanh-Nghi Doan, Thanh-Nghi Do, and François Poulet. 2013. Large Scale Image Classification with Many Classes, Multi-features and Very High-Dimensional Signatures. In *Advanced Computational Methods for Knowledge Engineering*. 105–116.

Thanh-Nghi Doan, Thanh-Nghi Do, and François Poulet. 2015. Large scale classifiers for visual classification tasks. *Multimedia Tools Appl.* 74, 4 (2015), 1199–1224.

Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. 1998. Inductive Learning Algorithms and Representations for Text Categorization. In *Proceedings of the Seventh International Conference on Information and Knowledge Management (CIKM '98)*. ACM, New York, NY, USA, 148–155.

Sebastiani Fabrizio. 2002. Machine Learning in Automated Text Categorization. *Comput. Surveys* 34 (2002), 1–47.

R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research* 9, 4 (2008), 1871–1874.

Y. Guermeur. 2007. SVM multiclasses, théorie et applications. (2007).

N. Japkowicz (Ed.). 2000. *AAAI'Workshop on Learning from Imbalanced Data Sets*. Number WS-00-05 in AAAI Tech Report.

Thorsten Joachims. 1998. Text categorization with Support Vector Machines: Learning with many relevant features. In *Machine Learning: ECML-98 (Lecture Notes in Computer Science)*, Claire Ndellec and Cline Rouveirol (Eds.). Springer Berlin Heidelberg, 137–142.

U. Kreßel. 1999. Pairwise classification and support vector machines. *Advances in Kernel Methods: Support Vector Learning* (1999), 255–268.

David D. Lewis and William A. Gale. 1994. A Sequential Algorithm for Training Text Classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94)*. Springer-Verlag New York, Inc., New York, NY, USA, 3–12.

Fei-Fei Li and Pietro Perona. 2005. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*. 524–531.

Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. 2009. Exploratory Undersampling for Class-Imbalance Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 39, 2 (2009), 539–550.

D.G. Lowe. 2004. Distinctive Image Features from Scale Invariant Keypoints. *International Journal of Computer Vision* (2004), 91–110.

Tien-Dung Mai, Thanh Duc Ngo, Duy-Dinh Le, Duc Anh Duong, Kiem Hoang, and Shinichi Satoh. 2017. Efficient large-scale multi-class image classification by learning balanced trees. *Computer Vision and Image Understanding* 156 (2017), 151–161.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. 2008. *Introduction to Information Retrieval* (1 ed.). Cambridge University Press.

Cam-Tu Nguyen, Xuan-Hieu Phan, and Thu-Trang Nguyen. 2010. JVnTextPro: A Java-based Vietnamese Text Processing Tool. (2010). http://jvntextpro.sourceforge.net.

OpenMP Architecture Review Board. 2008. OpenMP Application Program Interface Version 3.0. (2008).

Ioannis Partalas, Aris Kosmopoulos, Nicolas Baskiotis, Thierry Artières, George Paliouras, Éric Gaussier, Ion Androutsopoulos, Massih-Reza Amini, and Patrick Gallinari. 2015. LSHTC: A Benchmark for Large-Scale Text Classification. *CoRR* abs/1503.08581 (2015).

Maria Teresa Ricamato, Claudio Marrocco, and Francesco Tortorella. 2008. MCS-based balancing techniques for skewed classes: An empirical comparison. In *ICPR*. 1–4.

G. Salton, A. Wong, and C. S. Yang. 1975. A Vector Space Model for Automatic Indexing. *Commun. ACM* 18, 11 (Nov. 1975), 613–620.

S. Shalev-Shwartz, Y. Singer, and N. Srebro. 2007. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proceedings of the Twenty-Fourth International Conference Machine Learning*. ACM, 807–814.

Josef Sivic and Andrew Zisserman. 2003. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*. 1470–1477.

V. Vapnik. 2000. *The Nature of Statistical Learning Theory*. Springer-Verlag; 2nd edition.

Andrea Vedaldi and Brian Fulkerson. 2010. Vlfeat: an open and portable library of computer vision algorithms. In *Proceedings of the 18th International Conference on Multimedia 2010, Firenze, Italy, October 25-29, 2010*. 1469–1472.

Andrea Vedaldi and Andrew Zisserman. 2012. Efficient Additive Kernels via Explicit Feature Maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 3 (2012), 480–492.

S. Visa and A. Ralescu. 2005. Issues in Mining Imbalanced Data Sets - A Review Paper. In *Midwest Artificial Intelligence and Cognitive Science Conf.* Dayton, USA, 67–73.

G. M. Weiss and F. Provost. 2003. Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research* 19 (2003), 315–354.

J. Weston and C. Watkins. 1999. Support vector machines for multi-class pattern recognition. In *Proceedings of the Seventh European Symposium on Artificial Neural Networks*. 219–224.

Jianxin Wu. 2012. Power mean SVM for large scale visual classification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2344–2351.

Guo-Xun Yuan, Chia-Hua Ho, and Chih-Jen Lin. 2012. Recent Advances of Large-Scale Linear Classification. *Proc. IEEE* 100, 9 (2012), 2584–2603.

Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. 2010. Spark: Cluster Computing with Working Sets. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing (HotCloud'10)*. USENIX Association, 10–10.