A Distributed SVM Ensemble for Image Classification and Annotation

Nasullah Khalid Alham[1], Maozhen Li[1,2], Yang Liu[1], Mahesh Ponraj[1] and Man Qi[3]

[1]School of Engineering and Design, Brunel University, Uxbridge, UB8 3PH, UK
[2]The Key Laboratory of Embedded Systems and Service Computing, Ministry of Education, Tongji University, China
[3]Department of Computing, Canterbury Christ Church University, Canterbury, Kent, CT1 1QU, UK

Abstract— **Combination of classifiers leads to a substantial reduction of classification errors in a wide range of applications. Among them SVM ensembles with bagging have shown better performance in classification than a single SVM. However, the training process of SVM ensembles is notably computationally intensive especially when the number of replicated training datasets is large. This paper presents MRESVM, a MapReduce based distributed SVM ensemble algorithm for image annotation which re-samples the training dataset based on bootstrapping and trains SVM on each dataset in parallel using a cluster of computers. MRESVM is evaluated in a experimental environment and the results show that the MRESVM algorithm reduces the training time significantly while achieves high level of accuracy in classifications.**

Keywords-classificaton; SVM; ensemble classifiers; MapReduce

I. INTRODUCTION

Due to various complexities in classification problems, it is difficult to create classifiers with enhanced performance. The combination of classifiers leads to considerable reduction of misclassification errors in a wide range of applications. An ensemble classifier is generally superior in term of classification accuracy to a single classifier when the predictions of the base classifiers have sufficient error diversity [1]. Bagging [2] is the most commonly used combination approach which combines multiple classifiers by introducing randomness in the training instances. Bagging is useful in reducing the variance component of the expected misclassification error of a classifier [3]. Therefore bagging is effective particularly for classifiers with high variance and low bias, which are described in [2] as unstable classifiers. Unstable classifiers experience significant fluctuations with a small change of the training instances or other parameters [4].

SVM ensembles based on bagging have shown improved performance in classification compared with a single SVM [3] [5] [6] [7] [8]. Although some progress has been made by these approaches, current methods of bootstrapping create training datasets from the given training dataset by randomly re-sampling with replacement. Additionally ensemble learning is extremely computational intensive which limits their applications in real environments. Moreover SVM classifiers applied in ensemble learning require large computing resources due to the fact that computation time in SVM training is quadratic in terms of the number of training instances.

This paper presents MRESVM, a distributed SVM ensemble algorithm for automatic image annotation. MRESVM builds on the Sequent Minimal Optimization (SMO) algorithm [9] for high efficiency in training and employs MapReduce [10] for distributed computation across a cluster of computers. The MRESVM algorithm is designed based on the bagging architecture which trains multiple SVMs on bootstrap training datasets and combines the output in an appropriate manner. Each *map* function (called *mapper*) trains a SVM in parallel. The combination of SVMs is based on a 2 layered hierarchical structure that use second layer SVM to combine the first layer SVMs. The MRESVM algorithm reduces the training time significantly while keeping a high level of accuracy in classification compared with a single SVM.

The rest of this paper is organized as follow. Section II reviews some related work in SVM ensemble. Section III briefly introduces SVM ensemble techniques. Section IV describes in detail the design and implementation of the distributed MRESVM algorithm. Section V evaluates the performance of MRESVM in an experimental MapReduce environment. Section VI concludes the paper and points out some future work.

II. RELATED WORK

Currently ensemble methods represent one of the main research issues in machine learning. Mason et al. [11] show that ensembles enlarge the margins, consequently improve the generalization performance of learning algorithms. Schapire et al. [3] analyses ensemble learning methods based on bias variance decomposition of classification error which shows that ensemble classifiers reduce variance and bias, therefore reducing the overall classification error rate.

Bagging is the most commonly used method for constructing ensemble classifiers. Bagging introduces randomness in the training instances. Recently a number of SVM ensemble based on bagging have been proposed. Kim et al. [5] proposed SVM ensembles based bagging to improve the classification accuracy. The experimental results show improvement of classification accuracy of SVM ensemble. However, the experiments were performed with small training datasets. This approach of ensemble learning is extremely computational intensive for large training datasets which limits their applications in real environments. Yan et al. [6] presented a SVMs ensemble method based on bagging. The results show the ensemble method performs better than a single SVM. However, the algorithm is evaluated using a small number of bootstrap training datasets. Tao et al. [7] presented a SVM ensemble method based on bagging and random subspace to improve the user relevance feedback performance in content-based image retrieval. The results show improvement in classification accuracy. However the ensemble method cannot guarantee diversity within SVMs base classifiers due to the use of purely negative user feedback in the training process of SVMs.

Theoretical analysis of the bagging performance in terms of classification shows that expected misclassification error of bagging has the same bias component as a single bootstrap training dataset while the variance component is reduced significantly [3]. Valentini and Dietterich [12] presented a low bias SVM ensemble based on bagging. The aim is to reduce bias of the base SVMs before applying bagging. They consider the bias/variance tradeoffs to improve the classification accuracy of SVM ensemble. The experiments show improvement in classification accuracy. However, the algorithm was evaluated with small training datasets and the efficiency of the algorithm was not analyzed. This approach of ensemble learning is also extremely computational intensive for a large training dataset.

Lei et al. [13] proposed a SVM ensemble based on the bagging and boosting for text independent speaker recognition, and the experimental results show improvement of classification accuracy of SVM ensemble compared with single SVMs. However, this approach of ensemble learning is extremely computational intensive for large training datasets. Tang et al. [14] applied bootstrapping to create training datasets from the original training dataset. An SVM is trained on each dataset. The SVMs output are aggregated by Bayesian Sum Rule for a final decision. The algorithm is efficient and scalable. However there is a slight reduction in the accuracy level compare to standard SVM.

Summarizing, research on SVM ensemble algorithms has been carried out from various dimensions, but mainly focuses on improving classification accuracy [5] [6] [12]. Improving the efficiency of SVM ensemble in training still remains an open challenge. This motivates the design of MRESVM which is an efficient distributed SVM ensemble algorithm building on a highly scalable MapReduce implementation for image annotation.

III. SVM ENSEMBLE

A single SVM may not always provide a good classification performance on all test instances. To overcome this limitation, ensembles of SVMs have been proposed as a solution [3]. An ensemble of classifiers is a set of multiple classifiers combining a number of weak learners to create a strong learner. Training a diverse set of classifiers from a single training dataset has proven to be more accurate in classification than a single classifier [13]. There are a number of techniques for creating a diverse set of classifiers. The most common technique is to use re-sampling to diversify the training datasets based on Bootstrap Aggregating (bagging). Breiman [15] showed that bagging techniques can reduce the variance component of misclassification error, therefore increase the reliability of predictions. When the number of classifiers is large, the probability of error becomes small.

A two layered hierarchy approach is a combination method which uses a single SVM to aggregate the output results of a number of SVMs. Therefore, this method of combination consists of two layers of SVMs where the generated SVMs in the first layer are fed as input into a single SVM in the second layer [5]. Let $f_j(x),\ j = 1,2,3,...m$ be a decision function of the m^{th} SVM in the SVM ensemble and F be a decision function of SVM in the second layer. Then, the final decision of the SVM ensemble $f_{SVM}(x)$ for a given test instance x based on 2-layered hierarchical combining is determined by $f_{SVM}(x) = F(f_1(x), f_2(x),..... f_m(x))$, m is the number of SVMs in the SVM ensemble.

IV. DESIGN OF MRESVM

The MRESVM algorithm is based on the bagging architecture which trains multiple SVMs on bootstrap training datasets. Both random sampling with replacement and balanced sampling have been used.

As an initial step training datasets to train the base classifiers are created. For random sampling with replacement, m training datasets of size n are generated according to the uniform probability distribution from a dataset.

Each *map* task optimizes a training dataset in parallel in the first layer. The number of *map* tasks is equal to the number of training datasets. The output of each *map* task is the a_i array (Lagrange multipliers) for a training dataset and the training instances X_i which corresponds to Lagrange multipliers $a_i > 0$ in order to create input for the second layer. The output of the second layer includes the a_i array, bias threshold b and the training instances X_i which corresponds to $a_i > 0$ in order to calculate the SVM output u using equation;

$$u = \sum_{i=1}^{n} y_i a_i K(X_i, X) + b \qquad (1)$$

where X is an instance to be classified, y_i is class label for X_i and K is the kernel function. Each *map* task processes the associated instances datasets and generates a set of support vectors. Each set of support vectors is then combined and forwarded to the *map* task in the second layer as training instances. In the second layer a single set of support vectors is computed and the generated SVM model will be used in the classification. Figure 1 shows the architecture of MRESVM;

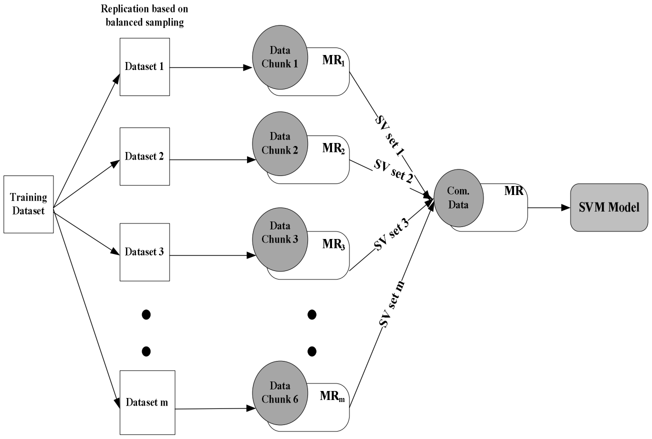

Figure1. MRESVM architecture

V EXPERIMENTAL RESULTS

We have incorporated MRESVM into our image annotation system which is developed using the Java programming language and the Weka package [16]. The image annotation system classifies visual features into pre-defined classes. Figure 2 shows the architecture of the system.

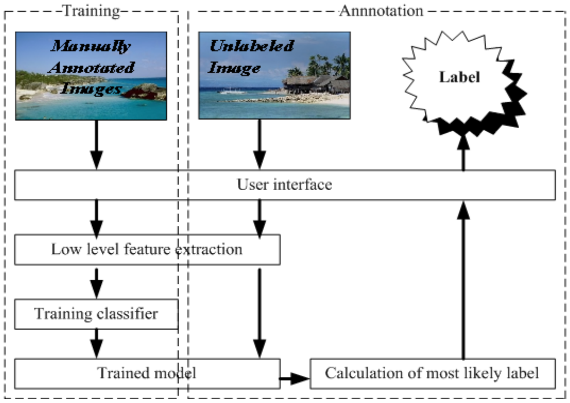
Training
Annnotation
Manually Annotated Images
Unlabeled Image
Label
User interface
Low level feature extraction
Training classifier
Trained model
Calculation of most likely label

Figure 2. Architecture of the image annotation system

Images are first segmented into blocks. Then, the low-level features are extracted from the segmented image blocks. Each segmented block is represented by feature vectors. We assign the low-level feature vectors to pre-defined categories. The system learns the correspondence between low level visual features and image labels. The annotation system combines low-level MPEG-7 descriptors such as scalable colour and edge histogram [17]. In the training stage, the SVM classifier is fed with a set of training images in the form of attribute vectors with the associated labels. After a SVM model is trained, it is able to classify a new image into one of the learned class labels in the training datasets.

A. Performance Evaluation

MRESVM is implemented using Weka's base machine learning libraries written in the Java programming language and tested in a Hadoop cluster. To evaluate MRESVM, the SMO algorithm provided in the Weka package is extended, configured and packaged as a basic MapReduce job. The Hadoop cluster for this set of experiments consist of a total of 12 physical cores across 3 computer nodes as shown in Table 1.

TABLE I. HADOOP CLUSTER CONFIGURATION

Hardware environment			
	CPU	Number of Cores	RAM
Node 1	Intel Quad Core	4	4GB
Node 2	Intel Quad Core	4	4GB
Node 3	Intel Quad Core	4	4GB
	Software environment		
SVM	WEKA 3.6.0 (SMO)		
OS	Fedora10		
Hadoop	Hadoop 0.20		
Java	JDK 1.6		

The performance of MRESVM is evaluated from the aspects of efficiency and accuracy. Figure 3 shows the speedup of the MRESVM in SVM training which achieves close to 12 times in speedup compared to the standalone SVM ensemble (SVM Ensemble).

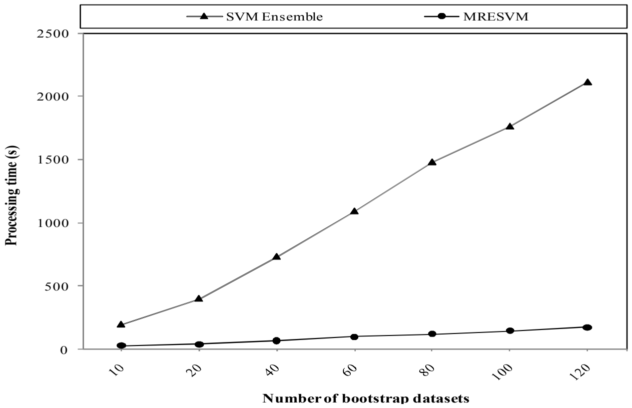

Figure 3. The speedup of MRESVM using 12 *Map* tasks

MRESVM outperform the standalone SVM ensemble with an increasing number of bootstrap training datasets in terms of training time required. Figure 4 shows the increasing efficiency with the number of participating MapReduce *mappers* varying from 4 to 12.

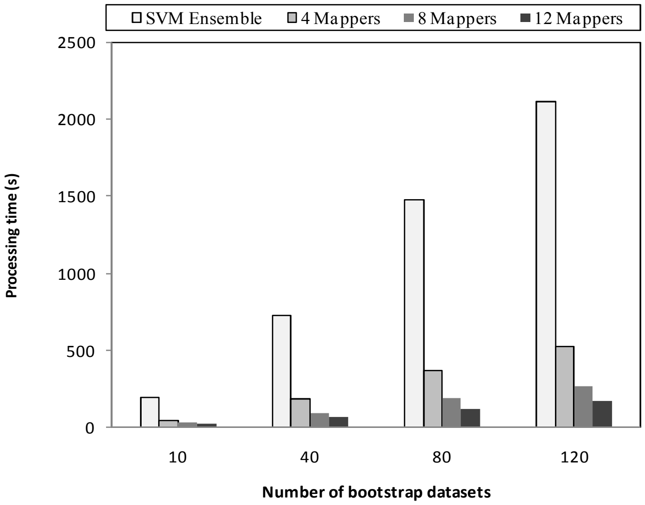

Figure 4. The overhead of MRESVM

Furthermore the accuracy of MRESVM with different sampling strategies was evaluated using 2000 training instances and the results are presented in Table 2. In total 250 test instances were used (10 test instances at a time), the average accuracy level was considered. The results show that MRESVM 100 bootstrap training datasets achieves up to 96% in accuracy which is higher than with single SVM.

TABLE II. SUMMARISED ACCURACY RESULTS

	Single SVM	MRESVM
Correctly Classified	≈ 94 %	≈ 96 %
Incorrectly Classified	≈ 6%	≈ 4 %

VI CONCLUSIONS

In this paper we have presented MRESVM, a scalable distributed SVM ensemble algorithm that capitalizes on the scalability, parallelism and resilience of MapReduce for large scale image annotations. By re-sampling the training dataset based on bootstrapping and training SVM on each training dataset in parallel using a cluster of computers, MRESVM is evaluated in an experimental environment showing that the distributed SVM algorithm reduces the training time significantly and achieves high level of accuracy in classifications.

A remarkable characteristic of the MapReduce Hadoop framework is its support for heterogeneous computing environments. Therefore computing nodes with varied processing capabilities can be utilized to run MapReduce applications in parallel. However, current implementation of Hadoop only employs first-in-first-out and fair scheduling with no support for load balancing taking into consideration the varied resources of computers. A future work will be to design load balancing schemes to optimize the performance of MRESVM in heterogeneous computing environments.

REFERENCES

[1] G. Brown, J.Wyatt, R. Harris, X.Yao, Diversity creation methods: a survey and categorization. Information Fusion, 6, 5-20, 2005.

[2] L. Breiman, Bagging predictors. Machine Learning, 24,123–140,1996.

[3] R. Schapire, Y.Freund, P. Bartlett, W. Lee, Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26, 1651–1686, 1998.

[4] G. Fumera, R.Roli, A. Serrau, Dynamics of Variance Reduction in Bagging and Other Techniques Based on Randomisation. Multiple Classifier Systems, pp. 316-325, 2005.

[5] H. Kim, S. Pang, H. Je, D. Kim, S. Bang, Support Vector Machine Ensemble with Bagging. SVM, pp. 397-407, 2002.

[6] G. Yan, G. Ma, L. Zhu, Support vector machines ensemble based on fuzzy integral for classification, in: ISNN, pp. 974–980, 2006.

[7] D. Tao, X. Tang, X. Li, X. Wu, Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28, 1088– 1099, 2006.

[8] N. Stepenosky, D. Green, J. Kounios, C. Clark, R. Polikar, Majority vote and decision template based ensemble classifiers trained on event related potentials for early diagnosis of Alzheimer's disease, IIEEE International Conference. Acoustic, Speech and Signal Proceedings, pp. 901-904, 2006.

[9] C. Platt, Sequential minimal optimization: a fast algorithm for training support vector machines, Technical Report, MSR-TR-98-14, Microsoft Research, 1998.

[10] R. Lämmel, Google's MapReduce programming model —revisited, Science of Computer Programming, 70, 1-30, 2008.

[11] L. Mason, P. Bartlett, J. Baxter, Improved Generalization through Explicit Optimization of Margins. Machine Learning, 38, 243-255, 2000.

[12] G. Valentini, T. Dietterich, Low Bias Bagged Support Vector Machines, in: ICML, pp. 752-759, 2003.

[13] Z. Lei, Y. Yang, Z. Wu, Ensemble of Support Vector Machine for Text-Independent Speaker Recognition, International Journal Computer Science and Network Security, 6,163-167, 2006.

[14] Y. Tang, Y. He, , S. Krasser, Highly Scalable SVM Modeling with Random Granulation for Spam Sender Detection, in: ICMLA, pp. 659-664, 2008.

[15] L. Breiman, Bias, variance and arcing classifiers, Technical Report TR 460, Statistics Department, University of California, Berkeley, CA 1996.

[16] Weka 3, [Online]: http://www.cs.waikato.ac.nz/ml/weka/ (Last accessed: 3 April 2011).

[17] T. Sikora, The MPEG-7 visual standard for content description-an overview, IEEE Transactions on Circuits and Systems for Video Technology, 11, 696-702, 2001.