Comments on "A Parallel Mixture of SVMs for Very Large Scale Problems"

Xiaomei Liu, Lawrence O. Hall[2], Kevin W. Bowyer
Department of Computer Science and Engineering
University of Notre Dame
South Bend, IN 46556
[2]Department of Computer Science and Eng., ENB118
University of South Florida
Tampa, FL 33620
hall@csee.usf.edu,{xliu5,kwb}@cse.nd.edu

January 7, 2004

Abstract

Collobert et. al. recently introduced a novel approach to using a neural network to provide a class prediction from an ensemble of support vector machines (SVMs). This approach has the advantage that the required computation scales well to very large data sets. Experiments on the Forest Cover data set show that this parallel mixture is more accurate than a single SVM, with 90.72% accuracy reported on an independent test set. While this accuracy is impressive, the referenced paper does not consider alternative types of classifiers. In this comment, we show that a simple ensemble of decision trees results in a higher accuracy, 94.75%, and is computationally efficient. This result is somewhat surprising and illustrates the general value of experimental comparisons using different types of classifiers.

1 Introduction

Support vector machines (SVMs) are most directly used for two-class classification problems (Vapnik, 1995). They have been shown to have high accuracy in a number of problem domains; for example, character recognition (Decoste, 2002). The training time required by support vector machines is an issue for large data sets. In (Collobert & Bengio, 2002), a method utilizing a mixture of support vector machines created in parallel was introduced to address the issue of scaling to large data sets. Results were reported on an example data set, the Forest Cover Type data set from the UC Irvine repository (Blake & Merz, 1998). The train data was converted to a two class problem. It was shown that the mixture of SVMs is more accurate than a single SVM, faster to train and has the potential to scale to large data sets. The accuracy from training on 100,000 examples is shown to be 90.72% on an unseen test set of 50,000 examples. The parallel mixture of SVMs was not compared with other classifiers.

In this comment, we compare an ensemble of randomized C4.5 decision trees (Dietterich, 1998) to the parallel mixture of SVMs and, perhaps contrary to our expectations and others, find that the ensemble of decision trees results in a more accurate classifier. Further, decision trees scale reasonably well with large data sets (Chawla, et.al., 2003). This result seems to reinforce the idea that is always useful to compare a classifier to other approaches.

In the next section, we briefly discuss the two ensemble classifiers compared. Section 3 provides the details of our experiments and our experimental results. Section 4 is the discussion and our conclusion.

2 Background

2.1 SVM and A parallel Mixture of SVMs.

The SVM was introduced by Vapnik (Vapnik, 1995). SVM classifiers are used to solve problems of two-class classification. The learning/training time for an SVM is high. It is at least quadratic in the number of the training patterns.

In order to decrease the time cost of SVM, Collobert (Collobert & Bengio, 2002) proposed a parallel mixture of SVMs. They only use part of the training set for each SVM, so the time cost is decreased significantly. It is conjectured that the time cost of a parallel mixture of SVMs is sub-quadratic with the number of training patterns for large scale problems. The performance of a parallel mixture of SVMs is claimed to be at least as good as a single SVM and shown to be so in (Collobert & Bengio, 2002).

2.2 Decision Trees

A decision tree (DT) is a tree-structured classifier. Each internal node of the decision tree contains a test on an attribute of the example to be classified, and the example is sent down a branch according to the attribute value. Each leaf node of the decision tree has the class value of the majority class for the training examples which ended up at the leaf. A DT typically builds axis-parallel class boundaries. Pruning is a useful method to decrease overfitting for individual DTs, but is not so useful for ensembles of decision trees.

The randomized C4.5 algorithm (Dietterich, 1998) was based on the C4.5 (Quinlan, 1993) algorithm. The main idea of randomized C4.5 is to modify the strategy of choosing a test at a node. When choosing an attribute and test at an internal node, C4.5 selects the best one based on the gain ratio. In the randomized C4.5 algorithm, m best splits are calculated (m is a positive constant with the default value 20), and one

of them is chosen randomly with a uniform probability distribution. When calculating the m best tests, it is not required that they be from different attributes. In an extreme situation, the m candidate tests may be from the same attribute.

3 The Forest Cover Type Data Set

The original forest cover type data set (Description, 2001) contains a total of 581,012 instances. For each instance, there are 54 features. There are seven class labels numbered from 1 to 7. The distribution of the seven classes is not even. Table 1 shows the class distribution.

Class Label	Meaning	Number of Records
1	Spruce/Fir	211840
2	Lodgepole Pine	283301
3	Ponderosa Pine	35754
4	Cottonwood/Willow	2747
5	Aspen	9493
6	Douglas-fir	17367
7	Krummholz	20510

Table 1: The Class Distribution of The Forest Cover Type Data Set (from (Description, 2001))

Since an SVM is most directly used for two-class classification, the original 7-class data set was transformed into a 2-class data set in the experiments of (Collobert & Bengio, 2002). The problem became to differentiate the majority class (class 2) from the other classes. Since we are going to compare the performance of a DT ensemble with their parallel mixture of SVMs, we use the same two class version of the forest cover type data set in our experiment.

We downloaded the data sets from (Data, 2003). They normalized the original forest cover data by dividing each of the 10 continuous features or attributes by the max-

imum value in the training set. There were 50,000 patterns in the testing set, 10,000 patterns in the validation set, and 400,000 patterns in the training set. We used the downloaded testing set and validation set as our testing set and validation set accordingly. However, we did not actually tune our ensemble based on the validation set. So, for us it serves as a second test set. We used the first 100,000 patterns in the downloaded training set as our training set. These are the exact data sets which were used in (Collobert & Bengio, 2002).

4 Experimental Results

We used the software USFC4.5 which is based on C4.5 release 8 (Quinlan, 1993) and modified by the researchers at University of South Florida (Eschrich, 2003), to do the DT experiments.

4.1 An Ensemble of 50 Randomized C4.5 Trees on The Full Training Set

Typically, a randomized C4.5 ensemble would consist of 200 decision trees. To compare with the 50 support vector machines, we restricted our ensemble to 50 decision trees. Each tree was built on the whole training set of size 100,000. Since there were no differences in the data set of each individual tree, we used randomized C4.5 to create each tree to generate a diverse set of trees for the ensemble (Banfield, 2003). A random choice from among the top 20 tests was used to build the trees. The trees in the ensemble were unpruned. The ensemble prediction was obtained by unweighted voting. So, the class with the most votes from individual classifiers was the prediction of the ensemble.

As shown in Table 2, the ensemble accuracy on the training data set was 99.81%, on the validation set was 94.85%, and on the testing set was 94.75%. We also list the minimum, maximum, and average accuracy of the 50 individual DTs included in the ensemble in Table 2. The test set accuracy compares favorably with the 90.72% accuracy of the parallel mixture of support vector machines.

	Ensemble	Minimum	Maximum	Average
Training Set	99.81%	97.27%	97.73%	97.51%
Validation Set	94.85%	88.62%	89.92%	89.42%
Testing Set	94.75%	88.81%	89.66%	89.25%

Table 2: The Accuracy of Dietterich's Randomized C4.5 on The Forest Cover Type Data Set, 50 trees.

4.2 An ensemble of 100 C4.5 Trees on Half of the Training Set

To get an idea of how much the randomized C4.5 was helping the classification accuracy, we created an ensemble of 100 trees each built on one-half of the training data. Each tree was trained on a randomly selected 50,000 examples from the 100,000 example training set. It is not guaranteed that each instance appears exactly 50 times in the training sets of the 100 DTs.

Since each training data set is clearly unique, we built a standard C4.5 decision tree on them. The trees were not pruned. Each of our trees was built on 25 times more training data than the SVM. However, only 100,000 unique examples are used. Each tree can be built in one CPU minute.

The ensemble performance on the testing set was 92.76%, the minimum single tree performance is 86.23%, the maximum single tree performance is 87.60%, and the average single tree performance is 87.10%. So the SVM mixture was outperformed by

an ensemble of plain C4.5 DTs with each tree grown on 1/2 the training data of one of the SVMs.

5 Discussion & Conclusion

According to the results reported in (Collobert & Bengio, 2002), the best performance of their parallel mixture of SVMs (using 150 hidden units and 50 Support Vector Machines) on the training set was 94.09%, on the validation set was around 91% (estimated from Figure 4 in (Collobert & Bengio, 2002)), and on the testing set was 90.72%.

In our experiments, the ensemble of 50 randomized DTs had much better performance. As shown in Table 3, its accuracy on the training set was 99.81%, on the validation set was 94.85%, and on the testing set was 94.75%. We did build an ensemble of 200 trees, but the accuracies were only very slightly greater. So, the ensemble becomes good quite quickly.

	Randomized C4.5 DTs	Parallel Mixture of SVMs
Training Set	99.81%	94.09%
Validation Set	94.85%	91%
Testing Set	94.75%	90.72%

Table 3: The Comparison of Accuracy Between Randomized C4.5 DTs And A Parallel Mixture of SVMs on The Forest Cover Type Data Set

Each support vector machine in the ensemble of classifiers was built from a disjoint data set of 2000 examples. The high accuracy obtained from such small training data sets and the scalability of the algorithm are impressive. Each SVM classifier used significantly less than the 100,000 or 50,000 training examples for each decision tree in the ensemble. The accuracy of the decision tree ensemble with half the size of the data was 2% less than with all training examples. Clearly, the decision tree accuracy

will decline with less examples. However, below we show some timings that indicate a decision tree ensemble can likely be built in time comparable or less than the SVM ensemble.

As to the running time, according to (Collobert & Bengio, 2002), it needs 237 minutes when using 1 cpu, and 73 minutes when using 50 cpus. We ran our experiments on a single computer. The cpu time to create the ensemble of 50 random C4.5 DTs is approximately 108 minutes. We used a Pentium III system, each processor had 1 GHz clock speed. Since it is an ensemble, it could be created in parallel with each processor starting with a different random seed. The cpu time to build each tree is approximately two minutes. The parallel time would then be on the order of 2 minutes plus communication time.

Further, an ensemble of 100 trees each created on a randomly selected 50,000 examples was still 2% more accurate than the ensemble of support vector machines. Each of these trees could be built in approximately 1 minute of cpu time in parallel.

From our experiments, it is shown that for the 2-class forest cover type data set, an ensemble of DTs has very good predictive accuracy. This advantage does not only exist in the 2-class forest cover type data set. We also did some experiments on the original 7-class cover type data set using a single DT. The performance of a single DT is promising too. It is much better than that of a single feedforward back propagation neural network both in accuracy and in speed.

The comparative results provided here underscore the need to compare classifier results with other types of classifiers even when it seems the answer would be known a priori. For a given data set, most people would guess that a SVM would be much better than a decision tree. So, if one designs a classifier that is even better than a single support vector machine intuitively it seems unnecessary to compare with classical

approaches with known limits such as decision trees. We are certain that a parallel mixture of support vector machines will outperform decision trees on some training data sets, but not this one. As noted above, decision trees result in good classification accuracy on the forest cover data set. They are both faster to construct than support vector machines on this data set and more accurate.

Acknowlegements: This work was supported in part by the United States Department of Energy through the Sandia National Laboratories ASCI VIEWS Data Discovery Program, contract number DE-AC04-76DO00789 as well the United States Navy, Office of Naval Research, under grant number N00014-02-1-0266.

References

Banfield, R.E., Hall, L.O., Bowyer, K.W., and Kegelmeyer, W. P. (2003). A New Ensemble Diversity Measure Applied to Thinning Ensembles, Multiple Classifier Systems Conference, Surrey, UK, 306 - 316.

Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases [http://www.ics.uci.edu/ mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

Blackard, J.A. and Dean, D. J. (1999). Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. Computers and Electronics in Agriculture 24, 131-151.

Chawla, N.V., Moore, Jr., T.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P. and Springer, C. (2003). Distributed Learning with Bagging-Like Performance, Pattern Recognition Letters, 24 (1-3), 455-471.

Collobert, R., Bengio, S., and Bengio, Y. (2002). A Parallel Mixture of SVMs for Very Large Scale Problems, Neural Computation, Neural Computation 14, 1105-1114.

Decoste, D. and Scholkopf, B. (2002). Training invariant support vector machines, Machine Learning, 46, Issue 1-3, Pages 161-190.

Dietterich T.G. (1998). An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. Machine Learning 40(2):139-158.

Eschrich S. (2003). Learning from Less: A Distributed Method for Machine Learning, Ph.D. Dissertation, University of South Florida, May.

Vapnik, V. (1995). The Nature of Statistical Learning Theory. New York: Springer Verlag.

Quinlan, J.R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, Inc., Redwood City, CA.

Data location (2003). http://www.idiap.ch/ collober/forest.tgz

Description of data (2001).
http://ftp.ics.uci.edu/pub/machine-learning-databases/covtype/covtype.info