A Tree Decomposition Method for Solving Large-Scale SVM Problems

Fu Chang
Institute of Information Science
Academia Sinica
Taipei, Taiwan
fchang@iis.sinica.edu.tw

I. Introduction

We present here a new method for solving large-scale SVM problems. The presented method decomposes a large data set into a number of smaller ones and trains SVMs on each of them. Since this method uses a decision tree to decompose a data set, we refer to it as the *decision-tree support vector machine* (DTSVM) method. For data sets whose size can be handled by current non-linear SVM training techniques, DTSVM can speed up the training by a factor of thousands, and still achieve comparable test accuracy.

The role of the decision tree as a decomposition scheme can have the following benefits when dealing with large-scale SVM problems. First, the decision tree may decompose the data set so that certain decomposed regions become homogeneous; that is, they contain samples of the same labels. In the testing phase, when a data point flows to a homogeneous region, we simply classify it in terms of the common label of that region. This helps alleviate the burden of SVM training, which is only conducted in heterogeneous regions. In fact, our experiments revealed that, for certain data sets, more than 90% of the training samples reside in homogeneous regions; thus, the decision tree method saves an enormous amount of time when training SVMs. Random partition, on the other hand, cannot produce such an effect, since random pooling of a set of samples can hardly create a homogeneous data set due to the independent sampling operation.

Another benefit of using the decision tree is the convenience it provides when searching for all the relevant parameter values to maximize the solution's validation accuracy, which helps maintain good test accuracy. The goal of the DTSVM method is to attain comparable validation accuracy while consuming less time than training SVMs on the full data sets. To achieve our purpose, we found that it is important to control the size σ of the tree-decomposed regions as well as the SVM-parameter values. For some data sets, σ could be set to 1,500, while for other data sets, it had to be set to a larger value. Thus, the DTSVM method makes σ an additional parameter to the usual SVM-parameters. Other decomposition methods do not attempt to search for the optimal size of decomposed regions. Such searches are particularly easy under the DTSVM method because a decision tree is constructed in a recursive manner; hence, obtaining a tree with a larger size of σ does not require the reconstruction of a decision tree corresponding to that size of σ.

Using a decision tree also helps alleviate the cost of searching for the optimal values of SVM-parameters. Searching for these values is important, but it takes a tremendous amount of time, especially when training non-linear SVMs. To the best of our knowledge, no other data-reduction methods have attempted to reduce the cost of this operation. Our strategy involves training SVMs with all combinations of SVM-parameter values *only* for decomposed regions with an initial σ-level. The optimal values of the SVM-parameters obtained at this level are not necessarily the same as those obtained at higher levels. However, we observe that the best values for a higher level are usually among the top-ranked values for the initial level. Therefore, when we want to train SVMs for a higher σ-level, we only train them with the top-ranked values obtained for the initial level. Given the n^p-complexity of SVM training, where $2 \le p \le 3$, conducting a full search of SVM-parameter values only in regions with the initial σ-level certainly reduces the SVM training time. In fact, our experiments show that such savings were possible even when the optimal σ-level was higher than the full size of the data set.

The experimental and theoretical aspects of DTSVM are described in [1]. The implementation of this method is available at
http://ocrwks11.iis.sinica.edu.tw/~dar/Download/WebPages/DTSVM.htm.

II. Evidence for the Efficiency of DTSVM

We demonstrate the efficiency of DTSVM in two types of experiments.

In the first type, there are seven medium-size data sets, whose detailed information is shown in Table 1. In these data sets, the largest number of samples is 494K and the largest number of feature is 62K. The methods that were compared with DTSVM on these data sets comprised CART [2], RDSVM, CBD [3], Bagging [4], LASVM [5], and LIBSVM [6]. RDSVM acts similar to DTSVM, except it decomposes a data set by a random partition.

To measure speedup factors, we took LIBSVM's training time as the baseline. The speedup factor of DTSVM on the medium-size data sets was between 4 and 3,691 for one-against-one (1A1) training, which were significantly higher than that of all the compared methods, except CART. However, CART achieved poor test accuracy rates in many data sets. The results are shown in Tables 2-4.

TABLE 1. The medium-size data sets used in our experiments.

Data set	No. of Labels	No. of Samples	No. of Features
Pen Hand Written (PHW)	10	10,992	16
Letter	26	20,000	16
Shuttle	7	58,200	9
Poker	10	25,010	10
Census Income (CI)	2	45,222	14
News20	20	19,927	62,060
KDD CUP 10% (KDD-10%)	5	494,021	41

TABLE 2. TRAINING TIMES OF THE SEVEN METHODS, EXPRESSED IN *SECONDS*. TRAINING TYPE = 1A1. CART, DTSVM and RDSVM outperformed the other methods.

	PHW	Letter	Shuttle	Poker	CI	News20	KDD-10%
CART	0.4	4.3	0.3	12.4	13.1	306.0	10.3
DTSVM	275	768	23	3,533	5,209	3,053	371
RDSVM	278	841	542	9,234	3,174	5,223	4,014
CBD	697	2,598	303	106,819	215,219	39,590	57,789
Bagging	2,204	7,575	2,100	7,979	6,100	35,176	17,123
LASVM	924	4,001	5,670	546,417	492,764	23,339	1,261,639
LIBSVM	1,192	4,157	5,096	1,307,667	315,130	27,071	1,369,600

TABLE 3. SPEEDUP FACTORS OF ALL THE SEVEN METHODS, EXCEPT LIBSVM THAT IS TAKEN AS THE BASELINE. TRAINING TYPE = 1A1. CART, DTSVM and RDSVM outperformed the other methods.

	PHW	Letter	Shuttle	Poker	CI	News20	KDD-10%
CART	3,320.3	974.4	19,157.9	105,143.3	24,066.7	88.5	133,009.6
DTSVM	4.3	5.4	221.6	370.1	60.5	8.9	3,691.6
RDSVM	4.3	4.9	9.4	141.6	99.3	5.2	341.2
CBD	1.7	1.6	16.8	12.2	1.5	0.7	23.7
Bagging	0.5	0.5	2.4	163.9	51.7	0.8	80.0
LASVM	1.3	1.0	0.9	2.4	0.6	1.2	1.1

TABLE 4. TEST ACCURACY RATES OF THE SEVEN METHODS. TRAINING TYPE = 1A1. CART performed poorly on several data sets; while CBD and Bagging lagged behind DTSVM on some data sets.

	PHW	Letter	Shuttle	Poker	CI	News20	KDD-10%
CART	95.51%	87.18%	99.95%	49.72%	80.97%	46.50%	99.95%
DTSVM	99.42%	97.60%	99.93%	57.56%	84.81%	83.22%	99.96%
RDSVM	99.10%	97.54%	99.61%	57.18%	83.48%	83.10%	99.43%
CBD	99.63%	95.25%	99.85%	56.75%	84.08%	75.23%	99.91%
Bagging	95.52%	93.09%	99.66%	56.29%	83.75%	76.55%	99.68%
LASVM	99.63%	97.54%	99.91%	56.70%	84.13%	83.10%	99.94%
LIBSVM	99.63%	97.54%	99.92%	56.75%	84.25%	83.10%	99.95%

In the second type of experiments, there were four large-size data sets, whose detailed information is shown in Table 5. In these data sets, the largest number of samples is 4.9M and the largest number of features is 16.6M. In all the four data sets, DTSVM could complete 1A1 training within 18.25 hours. The methods that were compared on large-size data sets included CART, LIBLINEAR [7] and DTSVM. The results are shown in Tables 6-7.

TABLE 5. THE LARGE-SIZE DATA SETS USED IN OUR EXPERIMENTS.

Data set	No. of Labels	No. of Samples	No. of Features
Forest	7	581,012	54
PPI	2	1,249,814	14
KDD-full	5	4,898,431	41
Webspam	2	240,000	16,609,143

TABLE 6. TRAINING TIME OF THE FOUR METHODS, EXPRESSED IN *SECONDS*. TRAINING TYPE = 1A1. CART and LIBLINEAR outperformed the other methods.

	Forest	PPI	KDD-full	Webspam
CART	1,912	23,217	320	29,332
LIBLINEAR	2,633	3,571	4,105	6,074
DTSVM	16,927	65,696	18,834	63,015
RDSVM	399,416	373,948	21,613	1,175,616

TABLE 7. TEST ACCURACY OF THE FOUR METHODS. TRAINING TYPE = 1A1. DTSVM outperformed, or performed comparably to, the other methods. CART performed rather well compared to LIBLINEAR.

	Forest	PPI	KDD-full	Webspam
CART	93.31%	88.11%	99.99%	98.44%
LIBLINEAR	72.79%	87.42%	99.87%	99.53%
DTSVM	94.61%	92.29%	99.99%	99.03%
RDSVM	77.46%	87.43%	99.72%	99.45%

III. SUMMARY

The experimental results on medium-size data sets demonstrated that DTSVM achieved significantly higher speedup factors than the compared methods that are designed to solve large-scale SVM problems. CART is not such a method, and achieved poor test accuracy on many data sets, despite its efficiency in training. DTSVM, on the other hand, achieved higher or comparable test accuracy to all the compared methods.

The results on large-size data sets also demonstrate the effectiveness and efficiency of DTSVM. In particular, comparing DTSVM with RDSVM shows that the effectiveness of DTSVM is not simply built on the decomposition of a data set, but also on using a decision tree as the decomposition scheme. Moreover, comparing the kernel-based DTSVM with the linear-based LIBLINEAR demonstrates the value of non-linear SVM as a classification method. One noteworthy result is that LIBLINEAR achieved the best test accuracy rate on "Webspam", presumably because a linear model fits this data set rather well. However, to verify this assumption, we need to compare the test accuracy rates of linear and non-linear models. DTSVM offers us an opportunity to make such a comparison.

REFERENCES

[1] F. Chang, C.-Y. Guo, X.-R. Lin, and C.-J. Lu, Tree decomposition for large-scale SVM Problems, *Journal of Machine Learning Research*, vol. 11, pp. 2855-2892, Oct 2010.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees. Chapman and Hall, 1984.

[3] N. Panda, E. Y. Chang, and G. Wu, "Concept boundary detection for speeding up SVMs," Proc. International Conference on Machine learning, pp. 681-688, 2006.

[4] L. Breiman. "Bagging predictors," Machine Learning, vol. 262, pp.123-140, 1996.

[5] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast kernel classifiers with online and active learning," Journal of Machine Learning Research, vol. 6, pp. 1579-1619, 2005.

[6] R.-E. Fan, P.-H. Chen, and C.-J. Lin, "Working set selection using second order information for training SVM," Journal of Machine Learning Research, vol. 6, pp. 1889-1918, 2005.

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," Journal of Machine Learning Research, vol. 9, pp.1871-1874, 2008.