Multi-Class Support Vector Machines – A Comparative Approach

E. Keshava Reddy and Jyothi Bellary

***Abstract*—Support Vector Machines (SVM) were originally designed for binary classification. Later it was extended to multi-class classification. Various methods have been proposed to construct a multi-class aclassifier by combining binary classifiers. As it is computationally more expensive to solve multi-class problems comparison of these methods using large scale problems have not been considered seriously. Especially for methods solving multi-class SVM in one step, a much larger optimization problem is required. So up to now the experiments are confined to small datasets. In this paper we give decomposition implementations for two such all-together methods. We then compare their performance with three methods based on binary classifications: "One- against-all" , "One-against-one" and DAG SVM. Our experiments indicate that one-against-one and DAG methods are suitable for practical use than other methods. Results also show that for large problems methods by considering all data at once in general need fewer support vectors.**

***Index Terms*—Support vector machines, multi-class classification, decompositon methods etc.**

I. Introduction

Support Vector Machines (SVM) was originally designed for binary classification. Later on it was extended to multi-class classification. In general there are two approaches for multi-class SVM. One is by constructing and combining several binary classifiers while the other is by directly considering all data in one optimization formulation. The formulation to solve multi-class SVM problems in one step has variables proportional to the number of classes. Therefore, for multi-class SVM methodsaa either several binary classifiers have to be constructed or a larger optimization problem is needed. Hence in general it is computationally more expensive to solve a multi-class problem than a binary problem with the same number of data. Experiments in this paper are confined to small data sets. In this paper we will give a decomposition implementation for two such "all-together" methods. Their performance is compared with three methods based on binary classification "one-against-all", "one-against-one","DAG SVM".

II. One-Against-All, One-Against-One, DAG SVM

The earliest used implementation or SVM multi-class classification is probably the one-against-all method. It constructs k SVM models where k is the number of classes.

Manuscript received April 15, 2012; revised May 27, 2012.

E. K. Reddy is with the Jawaharlal Nehru Technological University Anantapur, Anantapur.(e-mail: keshava_e@rediffmail.com)

J. Bellary is with the Aditya College Of Engineering, Madanapalle.(e-mail: jyothibellary@gmail.com)

The ith SVM is trained with all of examples in the ith class with positive labels, and all other examples with negative labels. Thus given l training data (x_1,y_1)- - - - (x_i,y_i) where $x_i \in R^n$, $i=1. \ldots l$ and $y_i \in \{1, \ldots, k\}$ is the class of x_i, the ith SVM solves the following problem:

$$\begin{aligned} &\min_{w^i b^i \xi^i} \tfrac{1}{2} (w^i)^T w^i + C \sum_{j=1}^{l} \xi_j^i \\ &(w^i)^T \Phi(x_j) + b^i \geq 1 - \xi_j^i, \text{ if } y_j = i, \\ &(w^i)^T \Phi(x_j) + b^i \leq -1 + \xi_j^i, \text{ if } y_j \neq i \\ &\xi_j^i \geq 0, j = 1 \ldots . l \end{aligned} \quad (1)$$

where the training data x_i are mapped to a higher dimensional space by the function Φ and C is the penalty parameter.

$\min_{w^i b^i \xi^i} \tfrac{1}{2} (w^i)^T$ means that we would like to maximize $2/\| w^i \|$, the margin between two groups of data. When data are not linear separable, there is a penalty term $C \sum_{j=1}^{l} \xi_j^i$ which can reduce the number of training errors. The basic concept behind SVM is to search for a balance between the regularization terms and the training errors. After solving (1) there are k decision functions: $(w^1)^T \Phi(x) + b^1 \cdots\cdots (w^k)^T \Phi(x) + b^k$.

We say x is in the class which has the largest value in the decision function:

$$\text{Class of } x \equiv \operatorname{argmax}_{i=1 \ldots k} ((w^i)^T \Phi(x) + b^i) \quad (2)$$

Practically we solve the dual of (1) whose number of variables is same as the number of data in (1). Hence k l-variable quadratic programming problems are solved.

Another method is one-against –one method. This method constructs k(k-1)/2 classifiers where each one is trained on data from two classes. For training data from i^{th} and j^{th} classes, we solve the following binary classification problem:

$$\begin{aligned} &\min_{w^{ij} b^{ij} \xi^{ij}} \tfrac{1}{2} (w^{ij})^T w^{ij} + C \sum_t \xi_t^{ij} \\ &\left((w^{ij})\right)^T \Phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, \text{ if } y_t = i, \\ &\left((w^{ij})\right)^T \Phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, \text{ if } y_t = j \\ &\xi_t^{ij} \geq 0, j = 1 \ldots . l \end{aligned} \quad (3)$$

There are different methods for testing after all k(k-1)/2 classifiers are constructed. After some tests voting strategy is adapted.Practically we solve the dual of (3) whose number of variables is same as the number of data in two classes. Hence if in average k each class has 1/k data points, we have to solve *k(k-1)/2* aquadratic programming problems where each of them has about 2l/k variables

Another method is DAGSVM, Directed Acyclic Graph Support Vector Machine. Its training phase is same as one-against-one method by solving *k(k-1)*/2 binary SVMs.

However in testing phase it uses a rooted binary directed acycliac graph which has *k(k-1)*/2 internal nodes and k leaves. Each node is a binary SVM of i^{th} and j^{th} classes. An advantage of using DAG is that some analysis of generation can be established. We have implemented all the three methods using LIBSVM.

III. A Method By Considering All Data At Once and A Decomposition Implementation

The idea is similar to one-against-all approach. It constructs k two-class rules where the m[th] function separates training vectors of the class m from other vectors.Hence there are k decision functions but all are obtained by solving one problem. The formulation is as follows:

$$\min_{w,b,\xi} \frac{1}{2}\sum_{m=1}^{k} w_{mw_m}^{t} \;^{j} + C \sum_{i=1}^{l} \sum_{m\neq yi} \xi_i^m$$

$$w_{yi}^T \Phi(x_i) + b_{yi} \geq w_m^T \Phi(x_i) + b_m + 2 - \xi_i^m, \text{ if } y_t = I \quad (4)$$

$$\xi_i^m \geq 0, i = 1 \ldots\ldots l\ , \ m \in \frac{\{1\ldots\ldots k\}}{yi}$$

And the decision function is same as that of one-against-all method.

Like binary SVM it is easier to solve the dual problem.

IV. Numerical Experiments

A. Data and Imaplementation

In this section we presented some experimental results on several problems from UCI Repository and Statlog collection of machine learning databases.From UCI repository we chose the following databases: Iris, wine, glass and vowel. From Statlog Collection we chose the following databases: Vehicle, Segment, dna, Satimage, letter and shuttle. Except for the problem dna, we scale all training data to be in [-1, 1]a. Then test data are adjusted using the same linear transformation. For the problem dna, we do not scale its binary attributes. The problem statistics are given as follows:

TABLE I: Problem Statistics

Problem	#training data	#testing data	#class	#attributes	statlog rate
iris	150	0	3	4	
wine	178	0	3	13	
glass	214	0	6	13	
vowel	528	0	11	10	
vehicle	846	0	4	18	
segment	2310	0	7	19	
dna	2000	1186	3	180	95.9
satimage	4435	2000	6	36	90.6
letter	15000	5000	26	16	93.6
shuttle	43500	14500	7	9	99.99

The most important criterion for evaluating the performance of these methods is their accuracy rate. However it is unfair to use only one parameter set and then compare these five methods. Practically for any method people find the best parameters by performing the model selection. This is conducted on the training data where the test data are assumed unknown. Then the best parameter set is used for constructing the model for future testing. To reduce the search space of parameter sets, here we train all datasets only with the RB kernel $K(x_i, x_j) \equiv e^{-\gamma\|x_i - x_j\|^2}$. In addition for these methods solving several binary SVMs (one-against-one, one-against-all, DAG), for each model we consider that C and C and γ of all binary problems are the same. Note that this issue does Not arise for two all-togeter methods as each model corresponds to only one optiamization problem. We use similar stropping criteria for all methods. For each problem we stop the optimization algorithm if the KKT violation is less than $10^{-3.}$ To be more precise, each dual problem of the one-against-one and one-against-all approaches has the following general term:

$$\min_{\alpha} f(\alpha)$$
$$y^T\alpha = 0$$
$$0 \leq \alpha_i \leq C$$

where $y_i = \pm$. Using similar derivation of the stopping criterion of the method by Crammer and Singer, we have $\max(\max_{\alpha_i \leq C, yi=1} -\nabla f(\alpha)_i, \max(\max_{\alpha_i > 0, yi=-1} -\nabla f(\alpha)_i) \leq \min(\min_{\alpha_i \leq C, yi=-1} \nabla f(\alpha)i, \min(\min_{\alpha_i > 0, yi=1} -\nabla f(\alpha)i) + 10^{-3}$

B. Results and Discussions

For each problem, we estimate the generalized accuracy using different kernel parameters γ and cost parameters C: γ = [2^4 , 2^3 , 2^2 , ... , 2^{-10}] and C =[2^{12} , 2^{11} , 2^{10} , ... , 2^{-2}]. Therefore for each problem we try 15 x 15 = 225 combinations. We use two criteria to estimate the generalized accuracy. For datasets dna, satimage, letter and shuttle where both training and testing sets are available, for each pait of (C, γ), the validation performance is measured by training 70% of the training set and testing the other 30% of the training set. Then we train the whole training set using the pair of (C, γ) that achieves the best validation rate and predict the test set. The resulting accuracy is presented in the "rate"a column of table II. Note that if several (C, γ) have the same accuracy in the validation stage, we apply all of them to the test data and report the highest rate. For the other six smaller datasets where test data may not be available, we simply conduct a 10-fold cross validation on the whole training data and report the best cross-validation rate.

TABLE II: A Comparison Using the Kernel (Best Rates Bold - Faced

	One-against-one		DAG		One-against-all		[10],[11]		C&S	
Problem	(C,γ)	rate	(C,γ)	rate	(C,γ)	rate	(C,γ)	rate	(C,γ)	rate
iris	$(2^{12},2^{-9})$	**97.333**	$(2^{12},2^{-8})$	96.667	$(2^{9},2^{-3})$	96.667	$(2^{12},2^{-8})$	**97.333**	$(2^{10},2^{-7})$	**97.333**
wine	$(2^{7},2^{-10})$	**99.438**	$(2^{6},2^{-9})$	98.876	$(2^{7},2^{-6})$	98.876	$(2^{0},2^{-2})$	98.876	$(2^{1},2^{-3})$	98.876
glass	$(2^{11},2^{-2})$	71.495	$(2^{12},2^{-3})$	**73.832**	$(2^{11},2^{-2})$	71.963	$(2^{9},2^{-4})$	71.028	$(2^{4},2^{1})$	71.963
vowel	$(2^{4},2^{0})$	**99.053**	$(2^{2},2^{2})$	98.674	$(2^{4},2^{1})$	98.485	$(2^{3},2^{0})$	98.485	$(2^{1},2^{3})$	98.674
vehicle	$(2^{9},2^{-3})$	86.643	$(2^{11},2^{-5})$	86.052	$(2^{11},2^{-4})$	**87.470**	$(2^{10},2^{-4})$	86.998	$(2^{9},2^{-4})$	86.761
segment	$(2^{6},2^{0})$	97.403	$(2^{11},2^{-3})$	97.359	$(2^{7},2^{0})$	97.532	$(2^{5},2^{0})$	**97.576**	$(2^{0},2^{3})$	97.316
dna	$(2^{3},2^{-6})$	95.447	$(2^{3},2^{-6})$	95.447	$(2^{2},2^{-6})$	95.784	$(2^{4},2^{-6})$	95.616	$(2^{1},2^{-6})$	**95.869**
satimage	$(2^{4},2^{0})$	91.3	$(2^{4},2^{0})$	91.25	$(2^{3},2^{1})$	91.7	$(2^{3},2^{0})$	91.25	$(2^{2},2^{2})$	**92.35**
letter	$(2^{4},2^{2})$	**97.98**	$(2^{4},2^{2})$	**97.98**	$(2^{3},2^{2})$	97.88	$(2^{1},2^{2})$	97.76	$(2^{3},2^{2})$	97.68
shuttle	$(2^{11},2^{3})$	99.924	$(2^{11},2^{3})$	99.924	$(2^{9},2^{4})$	99.910	$(2^{9},2^{4})$	99.910	$(2^{12},2^{4})$	**99.938**

Table II presents the results of comparing five methods. We present the optimal parameters. We present the optimal parameters (C, γ) and their accuracy rates. Note that C & S column means by Crammer and Singer method. It can be seen that the optimal parameters (C, γ) are in various ranges

for different problems so it is essential to test so many parameter sets. We also observe that their accuracyis very similar. That is , no one is statistically better than the others. Comparing to earlier results listed in Statlog, the accuracy obtained by SVM is competitive or even better. For example among the four problems, dna to shuttle, the one-against-one approach obtains better accuracy on satimage and letter. For the other two problems, the accuracy is also close to that in Table I

TABLE III: TRAINING TIME TESTING TIME, AND NUMBER OF SUPPORT VECTORS (TIME IN SECONDS; BEST TRAINING AND TEST TIME BOLD – FACED; LEAST NUMBER OF SVS ITALICIZED

	One-against-one		DAG		One-against-all		[10],[11]		C&S	
Problem	training	#SVs	training	#SVs	training	#SVs	training	#SVs	training	#SVs
	testing		testing		testing		testing		testing	
iris	**0.04**	16.9	**0.04**	*15.6*	0.10	16.0	0.15	16.2	16.84	27.8
wine	**0.12**	56.3	0.13	56.5	0.20	*29.2*	0.28	54.5	0.39	41.6
glass	**2.42**	*112.5*	2.85	114.2	10.00	129.0	7.94	124.1	7.60	143.3
vowel	**2.63**	345.3	3.98	365.1	9.28	392.6	14.05	*279.4*	20.54	391.0
vehicle	**19.73**	302.4	35.18	293.1	142.50	343.0	88.61	*264.2*	1141.76	264.9
segment	**17.10**	442.4	23.25	*266.8*	68.85	446.3	66.43	358.2	192.47	970.3
dna	**10.60**	967	10.74	967	23.47	1152	13.5	951	16.27	*945*
	6.91		**6.30**		8.43		6.91		6.39	
satimage	**24.85**	1611	25.1	1611	136.42	2170	48.21	*1426*	89.58	2670
	13.23		12.67		19.22		**11.89**		23.61	
letter	**298.08**	8931	298.62	8931	1831.80	10129	8786.20	7627	1227.12*	*6374*
	126.10		**92.8**		146.43		142.75		110.39	
shuttle	170.45	301	**168.87**	301	202.96	330	237.80	202	2205.78*	*198*
	6.99		5.09		5.99		4.64		**4.26**	

*: stopping tolerance $\epsilon = 0.1$ is used.

We also report the training time, testing time and the number of unique support vectors in table III. Note that they are results when solving the optimal model. For small problems there are no testing time as we conduct cross validation.

For training time, one-against-one and DAG methods are the best. In fact the two methods have the same training procedure. Though we have to train as many as k(k-1)/2 classifiers, as each problem is smaller(only data from two classes), the total training time is still less. Note that in table V.3 the training time of one-against-one and DAG methods may be quite different for the same problem (eg. Vehicle). This is due to the difference on the optimal parameter sets.

Comparing to table II, the difference on the best rates is apparent. The one-against-all method returns the worst accuracy for some problems. Overall one-against-all and DAG still perform well. The comparison on linear and nonlinear kernels also reveals the necessity of using non linear kernels in some situations. The observation that overall the RBF kernel produuces better accuracy is imaportant as otherwise we do not even need to study the decomposition methods which is specially designed for the nonlinear case. There are already effective methods to solve very large problems with the linear kernel.

TABLE IV: A COMPARISON USING THE LINEAR KERNEL (BEST RATES BOLD -FACED

	One-against-one		DAG		One-against-all		[10],[11]		C&S	
Problem	C	rate	C	rate	C	rate	C	rate	C	rate
iris	2^4	**97.333**	2^8	**97.333**	2^{12}	96.000	2^5	**97.333**	2^0	87.333
wine	2^{-2}	**99.438**	2^{-2}	98.315	2^2	98.876	2^{-1}	98.876	2^{-1}	99.438
glass	2^8	**66.355**	2^4	63.551	2^5	58.879	2^9	65.421	2^6	62.617
vowel	2^5	**82.954**	2^6	81.439	2^{11}	50.000	2^8	67.424	2^6	63.068
vehicle	2^5	80.615	2^5	**80.851**	2^{12}	78.132	2^{10}	80.142	2^4	79.669
segment	2^{12}	**96.017**	2^{11}	95.844	2^{12}	93.160	2^8	95.454	2^{-2}	92.165

Finally we would like to draw some remarks about the implementation of these methods. The training time of the one-against-all method can be further improved as now or each parameter set, k binary problems are atreated independently. That is, kernel elements used when solving one binary problem are not stored and passed to the other binary problems through they have the same kernel matrix. Hence the same kernel element may be calculated several times. However, we expect that even with such improvements it still cannot complete with one-against-one and DAG on the training time. For all other approaches, caches have been implemented so that all problems involved in one model can share them. On the otherhand, for all approaces, now different models (i.e. different parameter sets) are fully independent. There are no caches for passing kernel elements from one model to another.

V. CONCLUSION

We have discussed decomposition implementations for two all-together methods and compared them with three methods based on several binary classifiers: one-against-one,one-against-all,DAG. Experiments on large problems show that one-agianst-one and DAG may be more suitable for practical use. A future work is to test data with a very large number of classes. Especially people have suspected that there may have some more difference among these methods if the data set has few points in many classes.

REFERENCES

[1] C. L. Blake and C. J, Merz, "UCI repository of machine learning databases. Technical report, University of Caliornia," *Department of Inormation and Computer Science*, Irvine, CA, 1998.

[2] L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, L. Jackel, Y. L. Cun, U. Muller, E. Sackinger, and V. P. Vapnik et. al, "Comparison of classifier methods: a case study in handwriting digit recognition," *In International Conference on Pattern recogniton*, pp. 77-87. IEEE Computer Society Press, 1994.

[3] E. J. Brendensteiner and K. P. Bennett, "Multicategoty classification by support vector machines," *Computational Optimizations and Applications*, pp. 53-79, 1999.

[4] C. Cortes and V. Vapnik, "Support vector network," *Machine Learning*, vol. 20, pp. 273-297, 1995.

[5] K. Crammer and Y. Singer, "On the learnability and design of output codes for multiclass problems," *In Computational Learning Theory*, pp. 35-46, 2000.

[6] C. W. Hsu and C. J. Lin, "A simple decomposition method for support vector machines," *Machine Learning*, vol. 46, pp. 291-314, 2002.

[7] T. Joachims, "Making large-scale SVM learning practical," In C.J.C. Burges and et. al. *Advances in Kernel Methods – Support Vector Learning, Cambridge*, MA, MIT Press, 1998.

[8] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization. In C.J.C. Burges and A.J. Smola, editors," *Advances inKernel Methods – Support Vector Learning, Cambridge*, MA, MIT Press, 1998.

[9] E. Mayoraz and E. Alpaydin, "Support Vector Machines for multiclass classification," In IWANN, pp. 833-842, 1999.

[10] V. Vapnik, Statistical Learning Theory, Wiley, New York, NY, 1998.

[11] J. Weston and C. Watkins, "Multi-class support vector machines," In *M. Vereleysen, editor, proceedings of ESA NN99, Brussels*, D. Facto Press,1999.