ISSN:0975-9646

Anita Meshram et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (2) , 2015, 1660-1664

Advanced Probabilistic Binary Decision Tree Using SVM for large class problem

Anita Meshram[1] Roopam Gupta[2] and Sanjeev Sharma[3]

[1]School of Information Technology, UTD, RGPV, Bhopal, M.P., India.
[2]Information Technology, UIT, RGPV, Bhopal, M.P., India.
[3]School of Information Technology UTD, RGPV, Bhopal, M.P., India

Abstract -In this paper an algorithm of Advanced Probabilistic Binary Decision Tree (APBDT) using SVM for solving large classification problems is introduced, APBDT-SVM is tested in view of the size of the databases. APBDT-SVM integrates Binary Decision Tree (BDT) and Probabilistic SVM for solving multiclass classification issues. Probabilistic SVM uses standard SVM's output and sigmoid function to map the SVM output into probabilities. SVM's output merges with a sigmoid function, enlarge speed in decision making when combined with Binary Decision Tree (BDT). PSVM use to estimate the probability of membership to each sub-groups. APBDT-SVM lead to a dramatic improvement in recognition speed when addressing problems with large number of classes. Here Performance is evaluated in terms of classification accuracy, training and testing time by using standard UCI Machine Learning Repositories. The proposed APBDT-SVM method better performs for classification accuracy and computation time when compared to the other multiclass classification method like OaO, OaA, BDT and DAG.

Keywords
Support Vector Machine, Probabilistic SVM, Binary Decision Tree, separability measures

I. INTRODUCTION

Support vector machine (SVM) has become one of the most widely used machine learning rules, specifically for classification [1] [3] which was invented by Vapnik [2]. SVM is classifiers which is originally designed for solving binary classification problem and the extension of SVM to the multi-class problem is still an ongoing research issue [4]. The standard SVM multiclass approaches such as One-against-One (OaO), One-against-All (OaA) [5] or Directed Acyclic Graph (DAG) [6] have shown adequate result once separating the classes, but don't take into consideration the structure and the distribution of data. To overcome this drawback an easy and intuitive approach came which is based on building a binary decision tree [12].

On the basis of Binary Decision Tree and Probabilistic output of Support Vector Machine here present Advanced Probabilistic Binary Decision Tree (APBDT) using Support Vector Machine (SVM) as an original approach to the multi-class classification problem. This paper proposed Advanced Probabilistic Binary Decision Tree (APBDT) using SVM for solving large class problem. APBDT is an extension of Probabilistic Decision Tree [9]. Proposing Advanced Probabilistic Binary Decision Tree can be used in large class problem and it perform better when increases the size of the database. Instead of\ using a simple SVM classifier in each node, here propose PSVM classifier to estimate the probability of membership to each sub-group in the node. APBDT-SVM takes both the advantage of both the highest classification accuracy of SVM and the efficient computation of the tree architecture. The structure of decision Tree is constructed by measuring the distance between the gravity centers of different classes, An automated graph is generated where at each node a binary-class PSVM is trained.

For the readers' convenience, introduce the SVM briefly in section II. A brief introduction about the several widely used multiclass methods to divide the k classes in to binary class is in section III. The structure of the Binary Decision Tree is constructed by dividing the classes into groups and its subgroups. The complete description about the construction of decision tree is described in section IV. APBDT-SVM is the integration of SVM's output associate with sigmoid function, called Probabilistic SVM and Binary Decision Tree. Section V is a brief description about Probabilistic SVM. In section VI a brief description of APBDT-SVM methods and their algorithm is provided. Numerical experiments are explained in section VII which show that APBDT-SVM is suitable for practical use than other multiclass methods and it is also compatible for large class problem. section VIII gives a conclusion of the paper and future work.

II. SUPPORT VECTOR MACHINE:

The support vector machine (SVM) is a training algorithm for performing classification rule from the data set. SVM trains the classifier to predict the class of the new Sample. SVM is based on the concept of hyperplane that defines the decision boundary. The points that form the decision boundary between the classes is called support vector treated as a parameter. The principle of SVM is minimizing the structural risk in high dimensional feature space, search an optimal discriminant hyperplane with low dimension that separates two classes in a training set. Suppose we have l training set $\{(x_1, y_1), (x_2, y_2), \ldots\ldots (x_l, y_l)\}$ Where $x_i \epsilon R_d$ and $y_i \epsilon (+1, -1)$. Consider a hyperplane defined by (w, b), where w is weight vector and b is bias. The classification of new object x is done according to the decision function:

$$f(x) = sign(w.x + b) = sign(\sum_{i}^{N} \alpha_i y_i k(x_i.x) + b)$$

When feature space is nonlinear, then SVM maps nonlinear data into linear feature space by use of kernel functions and then find the optimal classification hyperplane in high dimensional feature space. Several kernel function that are used:

- Linear: $k(x_i, x) = x_i^T x_j$
- Polynomial: $k(x_i, x_j) = (\gamma x_i^T x_j + r)^d$
- Gaussian $k(x_i, x_j) = \exp(\frac{-\|x_i - x_j\|^2}{2\sigma^2})$
- Sigmoid: $\tanh(\gamma x_i^T x_j + r)$

In the APBDT - SVM approach we choose to use Gaussian kernel. These kernels have some attractive properties such as smoothness, feasible convolution formulae, and Fourier transforms. One important application is the high order extension of exact and accurate calculations. Gaussian kernel has a parameter σ^2, That small value will lead to curved hyper plans and high value will forced hyper plans to be straighter.

III. MULTICLASS CLASSIFICATION TECHNIQUE

SVM are generally designed for binary classification problem. Many researchers extend SVM in Multiclass classification problem. There are different methods in multiclass classification that solve the multiclass problem in SVM by dividing k number of classes into several binary sub-classes. Numerous methods for multiclass classifications are: One-against-All (OaA), One-against-One (OaO), Directed Acyclic Graph (DAG), Binary Decision Tree (BDT).

A. One against All (OaA):

OaA [5] [11] construct N binary-class SVMs. The i^{th} SVM are trained while i^{th} class is labeled by 1 and rest sample are labeled by -1. In the testing phase, a test example is presented to all N SVMs and is labelled according to the maximum output among the N classifier. The disadvantage of this method is that its training and testing phase are usually very slow.

B. One against One (OaO):

It constructs all possible N(N-1)/2 two class classifiers. Each classifier is trained by using the sample of first class is labeled 1 and sample of another class is labeled -1. To combine these classifiers a max-win algorithm is used. Each classifier casts one vote for its recommended class, and finally the class with the highest votes wins. This method, disadvantage is that, when the number of classes is large then OaO resulted slower testing because every test sample has been presented to the large number of N(N-1)/2 classifiers.

C. Directed Acyclic Graph (DAG):

The DAGSVM algorithm for training an N(N-1)/2 classifiers is same as OaO. In the testing part, the algorithm depends on rooted binary directed acyclic graph to make a decision. So the classification of the DAG is usually faster than OaO.

D. Binary Decision Tree (BDT):

BDT technique uses multiple SVMs arranged in a binary tree structure. SVM in each node of the tree is trained using two of the classes. In this architecture, N-1 SVM needed to be trained for N Class problem, but it only needs to test $log_2 N$ SVMs to classify a sample. This lead to an impressive improvement in recognition speed when addressing problems with big number of classes..

IV. CONSTRUCT BINARY DECISION TREE:

In APBDT-SVM, we use the BDT multiclass technique [12][13][14]. It is a simple and yet elegant approach. The BDT is based on recursive dividing the classes into two disjoint groups in every node of the decision tree. The structure of decision tree can be determined by measuring the separability between the classes. Euclidean, weighted Euclidean distance, Mahalanobis and Manhattan distance could be used for separability measures. We apply the Euclidean distance as the similarity measures between the class gravity center.

To build a binary decision tree, first start by dividing the classes into two disjoint group g1 and g2 as shown in Figure 1. This is done by calculating the N gravity center for N different classes and measuring the distance between the gravity center of all classes. Let the n_i denote the total number of i^{th} class patterns x_i where *i=1,2.........l*. The center point of i^{th} class is calculated using the following equation:

$$c_i = \frac{\sum_{m=1}^{n} x_m{}^i}{n_i}$$

Separability measures are used to calculate the distance between i^{th} class and j^{th} class patterns. Here Euclidean distance is used as separability measures. Then the Euclidean distance between the i^{th} class and j^{th} class patterns are calculated using the following equations:

$$d_{ij} = \sqrt{(c_i - c_j)^2}$$

The two classes, that have maximum Euclidean distance are assigned to each of the two groups. After that, the class with the minimum distance from one of the group assign to the corresponding group, it is done for all classes. The classes from the first group are assigned to left sub-tree and the classes to the second group are assigned to the right subtree. These processes are continuing by dividing each of the groups to its subgroup, until they achieved only one class per group which sets as a leaf in the decision tree.

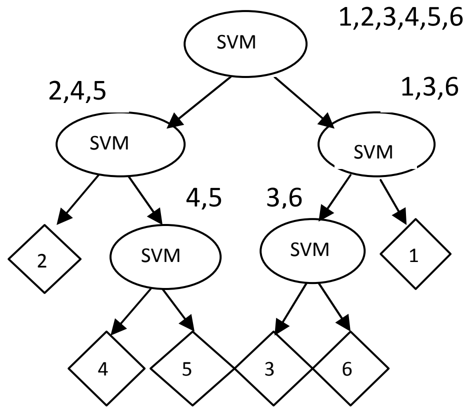

Figure 1: BDT for classification of 6 classes from 1 to 6

V. PROBABILISTIC SUPPORT VECTOR MACHINE:

SVM only gives a class prediction output that will be either +1 or -1. There are several approaches have been proposed in order to extract associated probabilities from SVM output. We will be focused on platt‘s[10] approach. He has composed a sigmoid function between the output f(x) of SVM and the probability of membership p(y=i|x) to a class i, specified by the attribute x. Sigmoid function can be expressed as:

$$p(y = 1|f(x) = \frac{1}{1 + e^{af(x)+b}}$$

Where a and b are parameters evaluated from the minimization of the negative log-likelihood function [6]

$$min \quad -\sum_i t_i \log p\big(f(x)\big) + (1 - t_i) \log(1 - p(f(x)))$$

And t_i is the new label of the classes. The probability of correct label can be deduced by applying the following formula. The estimate of target probability of positive and negative examples is,

$$t_+ = \frac{N_+ + 1}{N_+ + 2} \qquad t_- = \frac{1}{N_- + 2}$$

Where N_+ and N_- are the number of points that belongs to class *1* and class *2* respectively.

VI. ADVANCED PROBABILISTIC BINARY DECISION TREE USING SVM:

Advanced Probabilistic Binary Decision Tree using Support Vector Machine (APBDT-SVM) is new and an original approach to the multiclass classification problem. This method is a combination of the Binary Decision Tree (BDT) and Probabilistic output of SVM. In this algorithm SVM classifier associated with a sigmoid function (PSVM) is used to estimate the probability of membership to each sub-group in the node. APBDT-SVM takes the advantage of both the highest classification accuracy of SVM and the efficient computation of the tree architecture.

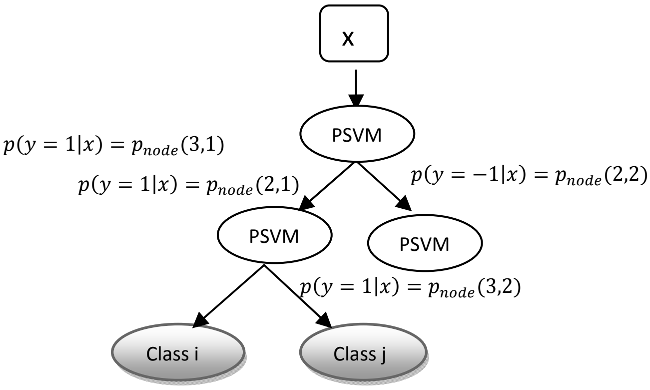

Figure 2: Advanced Probabilistic Binary Decision Tree using SVM

APBDT-SVM is based on recursively dividing the classes into two groups in every node of the Binary Decision Tree and training an SVM associate with sigmoid function i.e. Probabilistic SVM decides the assignment of the incoming unknown sample. The hierarchy of binary decision subtask as described above should be carefully designed before the training of each PSVM classifier. It is critical to have proper grouping for good performance of APBDT-SVM.

Here the unique probability function is employed for a trained tree, to get to a leaf node

$$p(y=i|x)=\prod_{h=1}^{leaf} p_{node}(h,l)$$

h is the level of the tree and $h=1$ is the root node. This expressly state that the probability of membership of an element to the class i is calculated as the product of the probabilities of the decisions adopted in all the nodes visited until arriving to the leaf. $node(h,l)$ means that l node in the h level. Once the tree builds, we will have the probability function, one for each class. When classifying unknown cases we would simply evaluate the probability function and then choose the class with the highest score.

A. Algorithm of Advance Probabilistic Binary Decision Tree using SVM:

We can understand the complete proposed work through these steps:

Step 1: Training phase

Inputs: Training set

Outputs: Probabilistic functions (one for each class)

1. Construct a binary decision tree.
   a. Calculate N gravity center for N different classes.
   b. Calculate Euclidean distance between two class's gravity centers.
   c. Let classes C_i and C_j have maximum Euclidean distance. The two classes which have maximum Euclidean distance assign in two different groups *g1* and *g2*.
   d. Classes which have minimum Euclidean distance with C_i Compared to C_j assign in *g1*, otherwise assign in *g2*.
   e. Go to next step if there is one class per group. Otherwise, repeat *step c* and *d*

2. Train an SVM classifier for each node of the decision tree.
   a. Calculate the hyper plane that separates the classes.
   b. If Separated classes are plural class, go to *step a*
   c. Else, go to the next step.
3. Fit sigmoid function to every SVM classifier trained in *Step 2*. Obtaining a probability function for the node.
4. For each leaf, probability function should be transverse all the corners.

End step 1

Step 2: Testing phase

Input: An unclassified example *x*

Output: Records of recommended classes and their corresponding probability

1. Estimate all the probability functions for the new unclassified example, using *step 1 (3-4)*.
2. Organize the classes corresponding to the probabilities.

End step 2

VII. EXPERIMENTAL EVALUATION:

On the basis of the above explained theory about the APBDT-SVM technique and different approaches for the multiclass classification, we perform the experimental work on these algorithms. These sections describe experimental works, datasets and obtained results on the algorithms. The classification algorithms were coded in standard MATLAB tool R2013a for performing the experiment. The operation is quantified by taking standard dataset from the UCI Machine Learning Repository [15] in different characteristics of Real, Categorical and Integer dataset having different number of instances, attribute and different number of classes.

Table 1: dataset description

datasets		#Attributes	#instances	Characteristics	#Classes
Small	Iris	4	150	Real	3
	Wine	13	178	Integer, Real	10
	Ecoli	8	336	Real	8
Medium	Movement _libra	91	460	Real	40
	Balance	4	625	Categorical	50
Large	Satellite	36	6435	Integer	100
	Thyroid	21	7200	Categorical, Real	110
	Yeast	8	1484	Real	120

Here separate the datasets into three categories small, medium and large, supported the number of classes and instances. These datasets are described in table 1. The dataset taken are Iris, wine and Ecoli in the small category, Movement Libra and balance in the medium category and at the last Yeast, Thyroid and Satellite in the large category based on number of classes. In our experiments, five different multiclass methods OaO, OaA, DAG, BDT, and APBDT were addressed. To determine the effectiveness of our proposed APBDT-SVM method, we compare the results obtained by the APBDT-SVM methods with OaO, OaA, DAG and BDT methods. Here SVM with Gaussian kernel are used for solving the binary classification problem.

Performance of classifiers is evaluated in terms of classification accuracy, training time and testing time on each data set. Table 2 through Table 4 depicts the outcomes of experiments employing by the data sets. Table 2 and Table 3 shows, training time and testing time of different multiclass methods, measured in second. Table 4 shows an accuracy result of the different multiclass method applied on each of the data sets.

Table 2: Training time of each multi class method (measured in seconds) for each dataset

Datasets/Methods		APBDT	BDT	OaO	OaA	DAG
Small	Iris	**2.212**	2.2753	2.4025	4.6325	3.9997
	Wine	**2.4723**	3.2422	3.4638	5.3451	4.6043
	Ecoli	**2.1100**	2.4265	3.7016	3.9667	4.1493
Medium	Movement _libra	**4.8658**	6.5041	15.4586	17.1634	34.9934
	Balance	**6.5562**	8.2383	22.8042	24.3039	60.4273
Large	Satellite	**7.7001**	9.0140	130.5940	156.1372	132.0593
	Thyroid	**8.5387**	15.8366	138.4242	162.5739	213.0549
	Yeast	**13.9598**	14.4727	198.4667	188.750	198.1867

Table 3: Testing times of each multi class method (measured in seconds) for every dataset

Datasets/Methods		APBDT	BDT	OaO	OaA	DAG
Small	Iris	**0.0417**	0.5199	2.8373	2.6727	0.5983
	Wine	**0.0891**	0.9683	4.3673	4.2308	0.9052
	Ecoli	**0.0429**	0.7316	4.1339	3.8434	0.5948
Medium	Movement _libra	**0.2148**	6.7031	15.3251	29.2360	13.8231
	Balance	**0.2259**	6.7936	19.9636	18.9350	22.1157
Large	Satellite	**0.1908**	9.6555	52.8051	60.8586	62.1716
	Thyroid	**0.1915**	17.8919	85.0812	68.5455	123.8939
	Yeast	**0.5565**	15.1811	76.3858	104.7796	124.9685

Table 4: Classification Accuracy (measured in %) of each multiclass method for every dataset

Datasets/Methods		APBDT	BDT	OaO	OaA	DAG
Small	Iris	**93.4947**	91.6667	84.1667	76.6667	93.3333
	Wine	**93.5421**	91.5000	82.5000	80	90.5000
	Ecoli	**93.4959**	72.7778	83.3333	84.4444	89.1111
Medium	Movement _libra	93.6678	82.6250	82.6250	85.5000	**94.5000**
	Balance	93.6789	67.3000	82.3000	82.4000	**94.9000**
Large	Satellite	**94.6438**	69.2500	71.4500	75.2500	89.5500
	Thyroid	**93.6445**	54.1818	65.6364	68.5909	87.9545
	Yeast	**94.0096**	56.7500	62.6250	59.1250	89.2917

In Table 1, 2 and 3, results and performance of the multiclass methods are shown. It is possible to observe that getting an improving result with large number of classes, especially with the large dataset satellite, yeast & thyroid. This means APBDT-SVM has the achieving result in training, testing and accuracy in all aspects. The table 4 shows the accuracy results where APBDT having maximum classification accuracy compared to the other multiclass classification method. The table 5 and figure 3 illustrates the mean of classification accuracy of the multiclass methods (OaO, OaA, DAG, BDT, APBDT).

Table 5: Mean Accuracy of all the multiclass methods

	APBDT	BDT	OAO	OAA	DAG
Mean Accuracy	93.7722	73.2564	76.8295	76.4971	91.1425

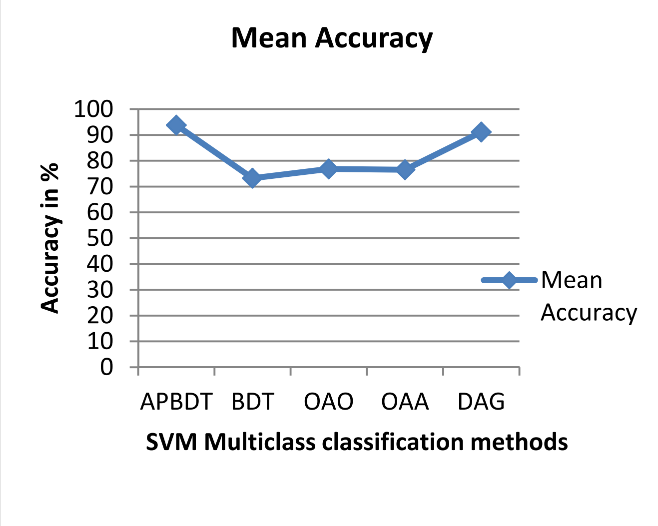

Figure 3: MEAN ACCURACY CHART

VIII. Conclusion and Future work:

The APBDT-SVM is providing better multiclass classification performance. Utilizing a decision tree architecture with a probabilistic output of SVM takes much less computation for deciding that in which class unknown sample is placed. Here proposed a new and original technique that combines Binary Decision Tree and SVM associate with a sigmoid function(PSVM) to estimate the probability of membership to each sub-groups. Probabilistic function for each leaf are built after traversing each nodes and leaves. It is critical to have proper structuring for the better performance of APBDT-SVM. After analyzing other multiclass methods like OaO, OaA, BDT and DAG we conclude that APBDT-SVM provides better classification accuracy. APBDT-SVM also provides better result in training and testing time compare to other multiclass methods. The result shows that APBDT is accurate and efficient method as an other multiclass method. APBDT-SVM lead to a dramatic improvement in recognition speed when addressing problems with maximum number of classes.

APBDT algorithm will be further used in various classifications of the data on various aspects. This algorithm will be applied in any application field where the performance of APBDT-SVM are tested in view of the size of the databases and the difficulties that it implies for processing them.

References:

[1] Z. N. Balagatabi, & H. N. Balagatabi,; "*Comparison of Decision Tree and SVM Methods in Classification of Researcher's Cognitive Styles in Academic Environment.*"; Indian Journal of Automation and Artificial Intelligence; Vol. 1, ISSN 2320-4001; 2013.

[2] V N. Vapnik; "*The nature of statistical learning theory*"; Springer, new York, 1995.

[3] R. Burbidge, B. Buxto., *An Introduction to Support Vector Machines for Data Mining*, Computer Science Dept., UCL, Gower Street, WC1E 6BT, UK; 2001.

[4] C. W. Hsu., C.J. Lin; "A *comparison methods for multiclass support vector machines*"; IEEE transaction on neural networks, vol. 13, no. 2, pg. 415-425, March 2002.

[5] Y. Liu and Y.F.Zheng; "*One-against-all multi-class SVM classification using reliability measures*"; IEEE international joint conference on neural network (IJCNN); vol. 2, pg: 849-854, 31 July-4 Aug. 2005.

[6] J. C. Platt, N. Cristianini and J. Shahere-Taylo;, "*Large margin DAGs for multiclass classification*"; Advances in neural information processing system, vol. 12, no. 3, pg. 547-553, 2000.

[7] G. madzarov, D. gjorgjevikj and I. chorbev; "*A multi-class SVM classifier utilizing binary decision tree*"; An international journal of

computing and informatics, Informatica; vol. 33 number 2; ISSN0350-5596, Slovenia; pg:233-241, 2009.

[8] B. E. Boser, I. M. Guyon, & V. N. Vapnik; A training algorithm for optimal margin classifiers. *5th Annual ACM Workshop on COLT*, pp. 144-152, 1992.

[9] J. S. Uribe, N. Mechbal, M. Rebillat, K. Bouamama, M. Pengov; "*Probabilistic Decision Tree using SVM for multiclass classification*"; Conference on control and fault-tolerant system. IEEE; pg. 619 to 624; Oct 9-11, France 2013.

[10] J. Platt; "*Probabilistic outputs for support vector machines and comparison to regularized likelihood methods*"; in Advances in large margin classifiers, Cambridge, MIT press, 2000.

[11] M. Arun Kumar, M. Gupta*; "Fast multiclass SVM classification using decision tree based one-against-all method"*; Springer, neural process lett, vol. 32, pg. 311-323; 25 Nov. 2010.

[12] G. madzarov, D. gjorgjevikj and I. chorbev*; "A multi-class SVM classifier utilizing binary decision tree"*; An international journal of computing and informatics, Informatica; vol. 33 number 2; ISSN0350-5596, Slovenia; pg:233-241, 2009.

[13] G. madzarov, D. gjorgjevikj; "*Evaluation of distance measures for multi-class classification in binary SVM decision tree"*; Springer link; Artificial intelligence and soft computing lectures notes in computer science; vol. 6113, pg. 437-444; 2010.

[14] G. madzarov, D. gjorgjevikj and I. chorbev*; "A multi-class SVM classifier utilizing binary decision tree"*; An international journal of computing and informatics, Informatica; vol. 33 number2; ISSN0350-5596, Slovenia; pg:233-241, 2009.

[15] K. Bache, & M. Lichman, *UCI Machine Learning Repository* [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science 2013.