



GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY  
INTERDISCIPLINARY

Volume 12 Issue 11 Version 1.0 Year 2012

Type: Double Blind Peer Reviewed International Research Journal

Publisher: Global Journals Inc. (USA)

Online ISSN: 0975-4172 & Print ISSN: 0975-4350

## Multiclass Classification and Support Vector Machine

By Yashima Ahuja & Sumit Kumar Yadav

*Lovely Professional University, Jalandhar (Punjab) India*

*Abstract* - In this paper we have studied the concept and need of Multiclass classification in scientific research. Various classification approaches are discussed in brief. Support Vector Machines (SVM) has well known record in Binary Classification. Our major emphasis in this paper is to study the fitness of Support Vector Machines in multiclass classification.

*GJCST-E Classification: G.4, C.1.2*



*Strictly as per the compliance and regulations of:*



# Multiclass Classification and Support Vector Machine

Yashima Ahuja<sup>α</sup> & Sumit Kumar Yadav<sup>σ</sup>

**Abstract** - In this paper we have studied the concept and need of Multiclass classification in scientific research. Various classification approaches are discussed in brief. Support Vector Machines (SVM) has well known record in Binary Classification. Our major emphasis in this paper is to study the fitness of Support Vector Machines in multiclass classification.

## I. INTRODUCTION

Multiclass classification is a major requirement in field of science and engineering because multiclass discrimination of objects is a serious problem in science and engineering. Multiclass classification is always considered complex than binary classification. In binary classification, only the decision boundaries of 1 class are to be known and rest (complement of first class) is considered as second class where as in multiclass classification, several boundaries are essential for that reason. This may lead to increase the probability of error because of constructions of many decision boundaries. Various classification methods like decision tree, KNN (k-nearest) are widely used for multiclass classification.

In this paper, our goal is to investigate the problem of multiclass classification and to propose an efficient method for the purpose. Support vector machines are highly accurate and able to model complex non-linear decision boundaries. The Major limitations of SVM are time complexity is very high. It leads to low-speed solutions and Svm are well known for binary classification.

To address the problem of multiclass classification, researchers have used 3 methods-

1. One vs one
2. One vs all
3. Directed acyclic graph

After analyzing the literature, we have found that DAG-SVM is considered to be best for multiclass classification.

## II. MOTIVATION

Multiclass problem can be expressed and solved as a single optimization problem, the following models proposed in the works by Suykens and

Vandewalle, Szedmak et al. and Oladunni & Trafalis were suggested.

Suykens and Vandewalle proposed a least square multi-support vector machine (LS-MSVM) for multi-category problems that can be interpreted as regularized least squares. It leads to a straightforward and rapid algorithm for generating linear or non-linear classifiers, obtained by solving a single linear system of equations. The approach also has alike interpretations as the reduced kernel multi-class machine. The drawback with LS-SVM is that sparseness is lost i.e. most of support vector becomes non-zero [13].

Szedmak et al. proposed a multiclass model for large sample sizes and number of features. In their formulation, the OAA framework is used, minimizing the L<sub>1</sub> norm of the normal vector w of the separating hyper planes. Their formulation solves k SVM optimization problems simultaneously and since there are no interactions between the variables of the k SVM problems, their method- which essentially is the OAA but considers all data at once produces the same solution as the separated k SVM problems with the L<sub>1</sub> norm in the objective function. Because their formulation considers all data at once, creating a large scale optimization problem, the size of the problem can be drawback if it gets too large [14].

In addition, Oladunni & Trafalis projected a regularized kernel multi-classification model by means of a reduced kernel approach. The resulting optimization problem solves for  $k(k-1)/2$  classifiers concurrently, and as there are no relations between the variables of the  $k(k-1)/2$  classification problems, the method can be considered a pair wise multiclass classification method that considers all data at one time. The formulation produces the similar solution as that of the independently solved  $k(k-1)/2$  binary classification problems. The method works well, however just like the SVM variant of Szedmak et al, the size of the problem can be a drawback. Consideration of all the data at once transforms the problem into a large scale optimization problem which can be very expensive to compute. Also, sparseness is lost [15].

The objective of this study is aimed towards the formulation of a piecewise kernel based multi-class model. The resulting optimization problem is linear programming problem with a linear cost function and

*Author α* : Research Scholar, Lovely Professional University, Jalandhar (Punjab) India. E-mail : er.yashima@yahoo.com

*Author σ* : Assistant Professor, Lovely Professional University, Jalandhar (Punjab) India.

linear constraints in dual space induced by a kernel function.

### III. SUPPORT VECTOR MACHINE

Machine Learning is ability to enable the computer to learn. It uses algorithm and techniques which perform different tasks and activities to provide efficient learning [3]. Our main problem is that how can we represent complex patterns and how to exclude bogus patterns. Support Vector Machine is a machine learning tool used for classification and regression. Support Vector Machine is based on supervised learning which classifies points to one of two disjoint half-spaces [2]. It uses nonlinear mapping to convert the original data into higher dimension. Its objective is to construct a function which will correctly predict the class to which the new point belongs and the old points belong. With an appropriate nonlinear mapping, two data sets can always be divided by hyperplane. Hyperplane separates the tuples of one class from another and defines decision boundary. There are many hyper planes that separate the data but only one will achieve maximum separation. The main reason behind maximum margin or separation because if we use a decision boundary to classify, it may end up nearer to one set of datasets compared to others [12]. This was the case when data is linear but mostly we find data is non-linear and data set is inseparable then we use kernels.

The core purpose of SVM is to separate the data with decision boundary and extends it to non-linear boundaries using kernel trick [12]. Major benefit of Svm is versatile means different Kernel functions can be specified for the decision function. General kernels are provided, but it is also possible to specify custom kernels. SVM becomes prominent when we used pixel maps as input; it gives accuracy equivalent to neural networks with elaborated features in a handwriting recognition task. Support vector machine is used for many applications such as text categorization, pattern recognition, face recognition, handwriting analysis but especially for classification and regression applications. Neural Networks are easier to apply than support vector machine but sometimes it provides insufficient results. Even the perceptron learning algorithms (e.g. gradient descent) are slower than SVM learning. Svm has been found to be unbeaten when used for pattern classification problems. One of the major challenge is that of choosing a suitable kernel for given application [12]. But there are many standard or default choices such as Gaussian or polynomial kernel but if these prove worthless then more elaborate kernels are needed.

Traditional Classification approaches perform weakly when working directly because of high dimensionality of data but support vector machine can

avoid the pitfalls of very high dimensionality representations. Support vector machine is the most promising technique and approach as compared to others. Support vector machine scales fairly well to high dimensional data and the trade-off between classifier complexity and error can be controlled explicitly. Another benefit of SVMs and kernel methods is that one can design and use a kernel for a particular problem that could be applied directly to the data without the need for a feature extraction process. It is particularly important in problems where a lot of structure of the data is lost by the feature extraction process. Example is text processing. Limitations of Svm are speed, size both in training and testing [8]. Discrete data presents another problem. Most severe difficulty with SVMs is the high algorithmic complexity and extensive memory requirements. In short we can say that the development of SVM is an utterly different from standard algorithms used for learning and SVM provides a fresh insight into this learning.

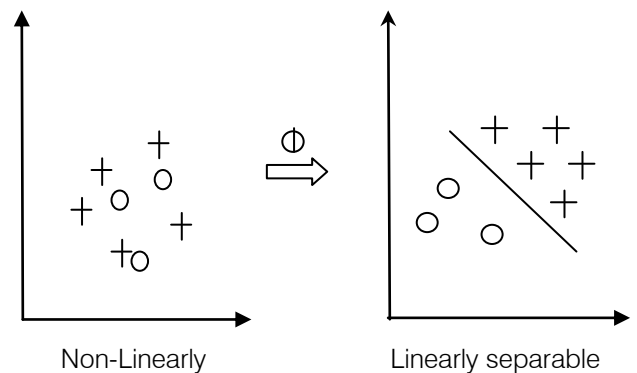
### IV. BINARY CLASSIFICATION USING SVM

For binary classification problems, the idea behind SVM is to split the data in finest method. Binary classification is used when we need to classify the two data sets. There are numerous examples of Binary classification like try-outs (one either makes or fails to make the team), claim size (large claims are above some threshold and small claims below), and fingerprint identification (matched or unmatched). Support vector machines are primarily designed for 2-class classification problems [2].

Support Vector Machine consider 2 approaches-

1. Case when the data are linearly separable
2. Case when the data are non-linearly separable

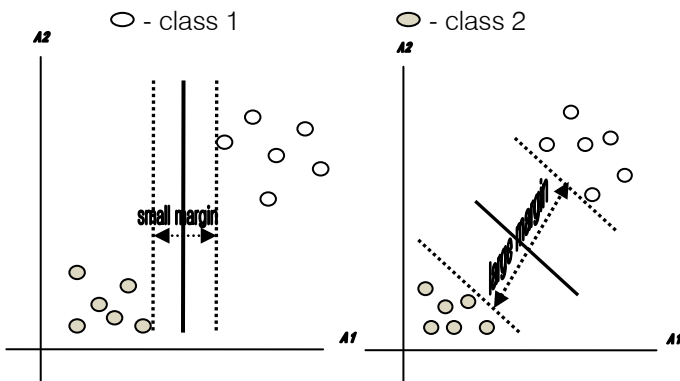
#### NON-LINEAR AND LINEAR SEPARATION



Let us consider first case; there are many linear decision boundaries that divide the data. But only one of these achieves maximum division. The main purpose we need it is because if we use a decision boundary to classify, it may end up nearer to one set of datasets compared to others and we do not want this to happen and thus concept of maximum margin classifier or hyper

plane as an apparent solution. Support vectors are the data points that lie closest to the decision surface. Support Vectors can be described as those data points that the margins pushes up against. They are the most difficult to classify [5]. The major problem here is to find the only optimal margin of the separating hyperplane  $W^T x + b = 0$ , the one that provides maximum margin between the classes. This margin guaranties the lowest rate of misclassification. The further advantage of margin would be avoiding local minima and better classification [12].

**DIAGRAM SMALL AND LARGE MARGIN SEPARATION**



To allow some flexibility in separating decision boundaries, SVM models have a cost parameter C that controls the tradeoff between allowing training errors and forcing strict margins. It creates a soft margin that permits some misclassifications.

Now, in second case data are not linearly separable i.e. in such cases no straight line can be found that would divide the classes. Linear svm's can be extended to generate nonlinear SVM'S for classification of linearly inseparable data. Such svm are capable of finding nonlinear decision boundaries.

**V. MULTICLASS CLASSIFICATION**

SVMs were mainly proposed to deal with binary classification but in today's life, we mostly have huge amount of data which we want to classify. Time series data represent quantities or trace the values taken by a variable over a period such as a month, year etc. Examples are stock market, price indexing etc. In this there will be more than two classes. So this creates the need of multiclass classification. Multiclass classification means classification with more than two classes.

Before introducing SVM, we have different kinds of multiclass techniques [9]. Firstly we will distinguish it on the basis of direct and indirect approach (via binary). Direct Approaches includes k-nearest neighbor, decision tree and bayes classification, linear classifications like perceptron. Multiclass classifications through binary include One-vs-one and One-vs-all, Directed acyclic graph svm, Error correcting output codes.

Nearest Neighbor classifiers are based on closeness. When given an unknown tuple, a k-nearest neighbor classifier searches the pattern space for the k training tuples that are closest to the unfamiliar tuple. The k training tuples are the k "nearest neighbors" of the unknown tuple [7]. Closeness is defined in terms of a distance metric such as Euclidean Distance [7]. Nearest Neighbor classifiers can be extremely slow when classifying test tuples [7]. It suffers from poor accuracy when given noisy or irrelevant attributes. Euclidean Distance can be calculated by -

$$d(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Decision Tree is a flowchart like structure where each internal node denotes a test on an attribute, every branch represents as a result of the test and each leaf node holds a class label. The topmost node in a tree is the root node. In classification, attribute values of the tuple are tested against the decision tree. Decision Trees can be easily converted into classification rules. Decision trees are popular because it doesn't require any domain knowledge, parameter setting and can handle multidimensional data with fast speed and good accuracy [1].

Bayes classification predicts class membership probabilities such as probability that a given tuple belongs to a particular class. It is based on bayes theorem. Bayes theorem provides a way of calculating posterior probability  $P(H / X)$  of H conditioned on X.

$$P(H / X) = P(X / H)P(H) / P(X)$$

Bayesian classifiers have the minimum error rate in comparison to all other classifiers but in practice this is not always the case sometimes inaccuracies in assumptions such as lack of available probability data [1].

Now let us consider the case Multiclass classification using Binary. In SVM, The idea of using a hyperplane to separate the data into two groups sounds well when there are only two target categories, but how does SVM handle case where the target variable has more than two categories or values? Numerous approaches have been suggested, but there are two most popular approaches described below [11].

In general, the most frequent method has been to construct one-versus-rest classifiers (usually referred to as "one-versus-all" or OVA classification) where each category is split out and all of the other categories are merged and to choose the class which classifies the test data with greatest margin. It divides an m class problem into m binary problems. The learning step of the classifiers is done by the whole training data,

considering the patterns from the particular class as positives and all other examples as negatives [3].

In validation phase, a pattern is presented to each one of the binary classifiers and then classifier which provides a positive output indicates the output class. In numerous cases, the positive outcome is not unique and some tie-breaking techniques are compulsory. The most familiar approach uses the confidence of the classifiers to decide the last outcome, predicting the class from the classifier with the maximum confidence. Rather than having a score matrix, when dealing with the outcomes of OVA classifiers (where  $r_i$  in  $[0, 1]$  is the confidence for class  $i$ ) a score vector is used:

$$R = (r_1, r_2, \dots, r_i, \dots, r_m)$$

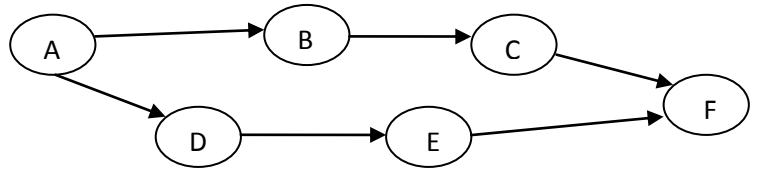
Another strategy is to construct a set of one-versus-one classifiers by dividing an  $m$  class problem into  $m(m-1)/2$  binary problems. Each problem comes up by a binary classifier which is responsible of distinguishing between a different pair of classes. The learning phase of the classifiers is done using as training data only a subset of instances from the original data-set which contain any of the two corresponding class labels, while the instances with dissimilar class labels are simply ignored [3].

In validation phase, a pattern is presented to each one of the binary classifiers. The output of a classifier given by  $r_{ij}$  in  $[0, 1]$  is the confidence of the binary classifier discriminating classes  $i$  and  $j$  in favor of the former class. Class with the largest confidence is the output class of a classifier. These outputs are represented by a score matrix  $R$ :

$$R = \begin{pmatrix} - & r_{12} & \dots & r_{1m} \\ r_{21} & - & \dots & r_{2m} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \dots & - \end{pmatrix}$$

There is so much difference between these two approaches. One-vs-all strategy constructs by fitting one classifier per class where as one-vs-one strategy constructs one classifier per pair of classes. Benefit of one-vs-all approach is its interpretability. This is the most commonly used strategy and is a fair default option. One-vs-one classifier doesn't scale well with  $n$  samples.

Directed Acyclic Graph can be used to represent a set of programs where the input, output, or execution of one or more programs is dependent on one or more other programs. The programs are nodes in the graph, and the edges (arcs) identify the dependencies [5]. A directed acyclic graph is a directed graph that contains no cycles.



## VI. CONCLUSION

In today's era, SVM are majorly used in binary class classification. In real time data, it is commonly found that data is to be classified in more than two classes. KNN, Decision Tree, Bayesian Classification are widely used for multiclass classification. KNN are lazy learners and hence perform slowly. Decision Trees are good for the purpose but highly complex if the depth of tree is on higher side. Bayesian Classifiers need good sample of data with high probable states, which is not always possible. In SVM, The idea of using a hyperplane to separate the data into two groups sounds well when there are only two target categories. For Multiclass Classification, One-vs-all strategy constructs by fitting one classifier per class where as one-vs-one strategy constructs one classifier per pair of classes. Benefit of one-vs-all approach is its interpretability. This is the most commonly used strategy and is a fair default option. One-vs-one classifier doesn't scale well with  $n$  samples.

## REFERENCES RÉFÉRENCES REFERENCIAS

1. Data Mining Concepts and Techniques book By Jiawei Han and Micheline Kamber
2. Wiki Svm, <http://www.dataminingtools.net/wiki/svm.php>
3. <http://www.dataminingtools.net/wiki/introml.php>
4. <http://sci2s.ugr.es/ovo-ova/>
5. <http://www.dtreg.com/svm.htm>
6. [http://research.cs.wisc.edu/condor/manual/v6.1/2\\_11Inter\\_job\\_Dependencies.html](http://research.cs.wisc.edu/condor/manual/v6.1/2_11Inter_job_Dependencies.html)
7. <http://alitarhini.wordpress.com/2011/02/26/parallel-k-nearest-neighbor/>
8. <http://www.svms.org/disadvantages.html>
9. <http://www.cs.berkeley.edu/~pliang/cs294-spring08/lectures/multiclass/slides.ppt>
10. Olutayo O. Oladunni and Gaurav Singhal, "Piecewise Multi-Classification Support Vector Machines", Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, June 14-19, 2009
11. Chih-Wei Hsu and Chih-Jen Lin, "A Comparison of Methods for Multiclass Support Vector Machines", IEEE Transactions on Neural Networks, Vol. 13, No. 2, March 2002
12. Laura Auria, Rouslan A. Moro, "Support Vector Machines (SVM) as a Technique for Solvency Analysis"

13. J.A.K. Suykens and J. Vandewalle, "Multiclass least squares support vector machine classifiers," in Proceedings of Joint Conf. On Neural Networks (IJCNN'99), Washington, D.C., 1999c.
14. S. Szedmak, J. Shawe-Taylor, C.J. Saunders, and D.R. Hardoon, "Multiclass classification by L1 norm support vector machine," in Pattern Recognition and Machine Learning in Computer Vision Workshop, Grenoble, France, May 02-04, 2004.
15. O.O. Oladunni and T.B. Trafalis, "A pair wise reduced kernel-based multi-classification Tikhonov regularization machine," in Proceedings of the International Joint Conference on Neural Networks (IJCNN '06), IEEE Press, pp. 130 - 137, Vancouver, BC, Canada, July 2006, on CD-ROM.





This page is intentionally left blank