International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 3 Issue: 12
ISSN: 2454-4248
118 – 125

Multi-class SVMs: A Relative Study of Kernels

Manju Bala
IP College for Women, Department of Computer Science, University of Delhi, Delhi
(e-mail: manjugpm@gmail.com)

ABSTRCT- Support Vector Machine is a powerful classification technique based on the idea of Structural risk minimization. Use of a kernel function enables the curse of dimensionality to be addressed. However, a proper kernel function for a certain problem is dependent on the specific dataset and till now there is no good method on how to choose a kernel function. In this paper, the choice of the kernel function was studied empirically and optimal results were achieved for multi-class SVM by combining several binary classifiers. The performance of the multi-class SVM is illustrated by extensive experimental results which indicate that with suitable kernel and parameters better classification accuracy can be achieved as compared to other methods. The experimental results of three datasets show that Gaussian kernel is not always the best choice to achieve high generalization of classifier although it often the default choice.

*****

I. INTRODUCTION

In the past two decades valuable work has been carried out in the area of text categorization [10],[11], optical character recognition [13], intrusion detection [14], speech recognition [18], handwritten digit recognition [20] etc. All such real-world applications are essentially multi-class classification problems. Multi-class classification is intrinsically harder than binary classification problem because the classification has to learn to construct a greater number of separation boundaries or relations. Classification error rate is greater in multi-class problem than that of binary as there can be error in determination of any one of the decision boundaries or relations.

There are basically two types of multi-class classification algorithms. The first type deals directly with multiple values in the target field *i.e.* K- Nearest Neighbor, Naive Bayes, classification trees in the class etc... Intuitively, these methods can be interpreted as trying to construct a conditional probability density for each class, then classifying by selecting the class with maximum a posteriori probability. For data with high dimensional input space and very few samples per class, it is very difficult to construct accurate densities. While the other approaches decompose the multi -class problem into a set of binary problems and then combining them to make a final multi-class classifier. This group contains support vector machines, boosting and more generally, any binary classifier. In certain settings the later approach results in better performance then the multiple target approaches.

Support Vector Machines (SVMs) originally designed for binary classification are based on statistical learning theory developed by Vapnik [5][19]. Larger and more complex classification problems have subsequently been solved with SVMs. How to effectively extend it for multi-class classification is still an ongoing research issue [16]. The most common way to build a $k-$class SVM is by constructing and combining several binary classifiers [9]. In designing machine learning algorithms, it is often easier to first devise algorithms to distinguish between two classes.

SVMs are learning machines that transform the training vectors into a high-dimensional feature space, labeling each vector by its class. It classifies data by determining a set of support vectors, which are members of the set of training inputs that outline a hyperplane in feature space [19]. It is based on the idea of Structural risk minimization, which minimizes the generalization error. The number of free parameters used in the SVM depends on the margin that separates the data points and not on the number of input features. SVM provides a generic technique to fit the surface of the hyperplane to the data through the use of an appropriate kernel function. Use of a kernel function enables the curse of dimensionality to be addressed, and the solution implicitly contains support vectors that provide a description of the significant data for classification [17]. The most commonly kernel functions are polynomial, gaussian and sigmoidal functions. Although in literature, the default choice of kernel function for most of the applications is gaussian. In training a support vector machine we need to select kernel function and its parameters, and value of margin parameter C. The choice of kernel function and parameters to map dataset well in high dimension may depend on specific datasets. There is no method to determine how to choose a appropriate kernel function and its parameters for a given dataset to achieve high generalization of classifier. The main modeling freedom consists in the choice of the kernel function and the corresponding kernel parameters, which influences the speed of convergence and the quality of results. Furthermore, the choice of the regularization parameter C is vital to obtain good classification results.

In this paper, we have studied the choice of the kernel function empirically and optimal results were achieved for multi-class SVM using three benchmark datasets of UCI

ISSN: 2454-4248
118 – 125

repository of Machine Learning Databases [1]. This paper is organized as follows. Section 2 briefly reviews the basic theory of SVM. In section 3, we demonstrates the experiments of multi-class SVM (one against all) using different kernel functions on few datasets of UCI repository and provides the comparative result of classifier accuracy of multi-class SVM for different kernel functions. We also compare the accuracy with the available results for different datasets. We conclude and discuss scope of future work in section 4.

II. SUPPORT VECTOR MACHINES

A. Theory of Support Vector Machines

This section briefly introduces the theory of SVM. Let $\{(x_1, y_1),\ldots,(x_m, y_m)\} \in R^n \times\{+1,-1\}$ be a training set. The SVM classifier finds a canonical hyperplane $\{x \in R^n : w^T x + b = 0,\ w \in \mathcal{R}^n, b \in \mathcal{R}\}$, which maximally separates given two classes of training samples in R^n. The corresponding decision function $f : \mathcal{R}^n \rightarrow \{+1,-1\}$ is then given by $f(x) = \text{sgn}(w^T x + b)$. For many practical applications, the training set may not be linearly separable. In such cases, the optimal decision function is found by solving the following quadratic optimization problem:

Minimize: $$J(\mathbf{w}, \xi) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i$$

Subject to

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1-\xi_i,$$

$$\xi_i \geq 0, i = 1,2,\ldots,m \qquad (1)$$

where ξ_i is a slack variable introduced to relax the hard-margin constraints and the regularization constant $C > 0$ determines the trade-off between the empirical error and the complexity term. The generalized optimization is based on a theorem about the VC dimension of canonical hyperplanes. It was shown that if the hyperplane is constructed under the constraint $\|\mathbf{w}\| \leq A$ then the VC- dimension of the class H is bounded by $h \leq \min(R^2A^2, n)+1$ [19], where R is the radius of the smallest sphere around the data. Thus, if we bound the margin of a function class from below, say by $2/A$, we can control its VC dimension and hence apply the SRM principle.

Applying the Karush-Kuhn Tucker complimentarily condition [7] which gives optimal solution of a non-linear programming problem, we can write $\mathbf{w} = \sum_{i=1}^{m} y_i\alpha_i \mathbf{x}_i$ after minimizing (1). This is called the dual representation of $\mathbf{w}$. A $\mathbf{x}_i$ with nonzero α_i is called a support vector. The coefficients α_i can be found by solving the dual problem of (1):

Maximize: $$L(\boldsymbol{\alpha}) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\mathbf{x}_j$$

Subject to

$$0 \leq \alpha_i \leq C, i = 1,2,\ldots,l$$

and $$\sum_{i=1}^{m}\alpha_i y_i = 0$$

(2)

Let S be the index set of support vectors, then the optimal decision function becomes

$$f(\mathbf{x}) = sgn\left(\sum_{i\in S} y_i\,\alpha_i\,\mathbf{x}^T\,\mathbf{x}_i + b\right)$$

(3)

The above equation gives an optimal hyperplane in R^n. However, more complex decision surfaces can be generated by employing a nonlinear mapping $\Phi : \mathcal{R}^n \rightarrow F$ while at the same time controlling their complexity and solving the same optimization problem in F. It can be seen from (2) that $\mathbf{x}_i$ always appears in the form of inner product $\mathbf{x}_i^T\,\mathbf{x}_j$. This implies that there is no need to evaluate the nonlinear mapping Φ as long as we know the inner product in F for a given $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{R}^n$. So, instead of defining $\Phi : \mathcal{R}^n \rightarrow F$ explicitly, a function $K : \mathcal{R}^n \times \mathcal{R}^n \rightarrow \mathcal{R}$ is introduced to define an inner product in F. The only requirement on the kernel $K(\mathbf{x}, \mathbf{y})$ is to satisfy Mercer's condition, which states: There exists a mapping Φ and an expansion

$$K(\mathbf{x}, \mathbf{y}) = \sum_i \Phi(x)_i\,.\Phi(y)_i$$

if and only if, for any $g(\mathbf{x})$ such that

$$\int g(\mathbf{x})^2\,d\mathbf{x} \text{ is finite}$$

then

$$\int K(\mathbf{x}, \mathbf{y})\,g(\mathbf{x})\,g(\mathbf{y})\,d\mathbf{x}\,d\mathbf{y} \geq 0.$$

Substituting $K(\mathbf{x}_i, \mathbf{x}_j)$ for $\mathbf{x}_i^T\ \mathbf{x}_j$ in (3) produces a new optimization problem:

Maximize $$L(\boldsymbol{\alpha}) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

Subject to $$0 \leq \alpha_i \leq C, i = 1,\ldots..,m$$

and $$\sum_{i=1}^{m}\alpha_i y_i = 0$$

(4)

Solving it for $\boldsymbol{\alpha}$ gives a decision function of the form

$$f(x) = \text{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i K(x_i, x_j) + b\right)$$
(5)

Whose decision boundary is a hyperplane in F, and translates to nonlinear boundaries in the original space.

B. Multi-class Support Vector Machines

All real world classification problems often involve more than two classes. Therefore, binary SVMs' are usually not enough to solve the whole problem. The most common way to build a k-class SVM is by constructing and combining several binary classifiers. To solve multi-class classification problems, we divide the whole pattern into a number of binary classification problems. The two representative ensemble schemes are One against All (1-vs-many) and One against One (1-vs-1) [12]. One against All is also known as "one against others." It trains k binary classifiers, each of which separates one class from the other $(k-1)$ classes. Given a point X to classify, the binary classifier with the largest output determines the class of X. One against One constructs $k(k-1)/2$ binary classifiers. The outputs of the classifiers are aggregated to make the final decision. Decision tree formulation is a variant of One against All formulation based on decision tree. Error Correcting output code is general representation of One against All or One Vs One formulation, which uses error correcting codes for encoding outputs [dietterich].The One against All approach, provides better classification accuracy in comparison to others [16]. Consequently, we have applied One against All approach in our experiments.

Commonly used kernels for decision functions of a binary SVM classifier such as polynomial, Gaussian and sigmoid may not be suitable for binary classification to map every dataset well in high dimensional space. There can be other functions, which satisfy Mercer's condition and can enhance classifier accuracy by appropriate transformation in high dimensional space. Few kernel functions [2] used in our experiment are shown in Table I.

Table I: Kernel Functions

Kernel Function	$K(x, x_i)$ for $\gamma > 0$
Cauchy	$1/(1+\gamma\,|x-x_i|^2)$
Gaussian	$e^{-\gamma|x-x_i|^2}$
Hyperbolic Secant	$2/(\exp(\gamma\,|x-x_i|)+\exp(-\gamma\,|x-x_i|))$
Laplace	$\exp(-\gamma\,|x-x_i|)$
Squared Sinc	$\sin^2(\gamma\,|x-x_i|)/(\gamma\,|x-x_i|)^2$
Symmetric Triangle	$\max(1-\gamma\,|x-x_i|, 0)$

III. EXPERIMENTAL RESULTS

A. Dataset

In this section, we evaluated the performance of multi-class SVM using different kernel functions on the *Iris*, *Wine* and *Glass* datasets of UCI repository of Machine Learning [1]. The *iris* dataset records the physical dimensions of three different classes of Iris flowers. There are four attributes in the *Iris* dataset. The class Setosa is linearly separable from the other two classes, whereas the latter two are not linearly separable from each other. The *Wine* dataset was obtained from chemical analysis of wines produced in the same regions of Italy but derived from three different cultivars. There are 13 attributes and 178 patterns in this *wine* dataset. There are three classes corresponding to the three different cultivars. The collections of the *Glass* dataset were for the study of different types of glass, which was motivated by criminological investigations. At the scene of crime, the glass left can be used as evidence if it is correctly identified. The *Glass* dataset contains 214 cases. There are nine attributes and six classes in the *Glass* Dataset.

C Results

The generalization performance is evaluated via a ten-fold cross-validation for each dataset. We have considered *One against All* method for designing a multi-class SVM. For a *k*-class problem, we have developed multi-class SVM by combining *k*-binary SVM with the same value of C and γ and tested the performance for different choices of kernel functions on predefined datasets. The most important criterion for evaluating the performance of multi-class SVM is their accuracy rate. For each multi-class SVM for a given kernel function, the hyperparameters space is explored on a two dimensional grid with the following values: $\gamma = [2^{-10}, 2^{-9}, 2^{-8}, \ldots, 2^{4}]$ and $C = [2^{-2}, 2^{-1}, 2^{0}, \ldots, 2^{12}]$. For all the 255 possible combinations of C and γ, the best and the average cross-validation accuracy is computed and rounded up to three decimal places. All the experimental tests were performed on a computer having Pentium 4 processor with 512MB RAM.

 ISSN: 2454-4248
118 – 125

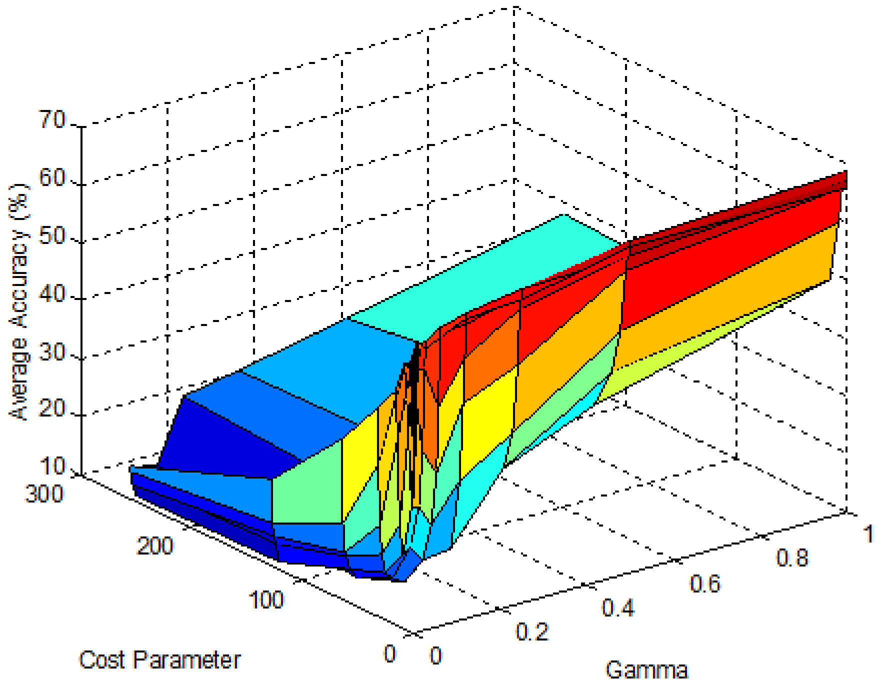


Fig. 1. Average classifier accuracy for Glass dataset using Gauss Kernel

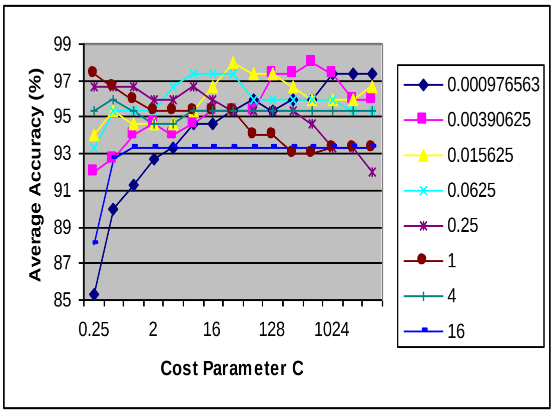
Fig. 1 shows the average cross-validation accuracy of the multi-class SVM classifier for the glass dataset using Gaussian kernel as a function of the two parameters C and γ. The figure below shows the variation in accuracy for 121 combinations of C and γ only. The optimal values of parameters can be chosen by visualizing the maximum value of average accuracy attained on the grid.

Similarly, the experiments were performed on different datasets using other kernel functions. We observed that multi-class SVM demonstrates better accuracy for certain value of C and γ. The significance of choosing appropriate values of C and γ can be realized from above 3D plot. However, we have analyzed the cross sectional view of the same in 2D. Due to the scarcity of space, we are presenting only the graphs for the kernel function having wider variation in classifier accuracy. Fig. 2(a) shows the variation of classifier accuracy of iris dataset with C for different values of γ using Gauss kernel function. The variation of classifier accuracy of wine dataset with C for different values of γ using Squared Sinc kernel function is shown in Fig. 2(b). Similarly, Fig. 2(c) shows the variation of classifier accuracy of glass dataset with C for different values of γ using Laplace kernel function.

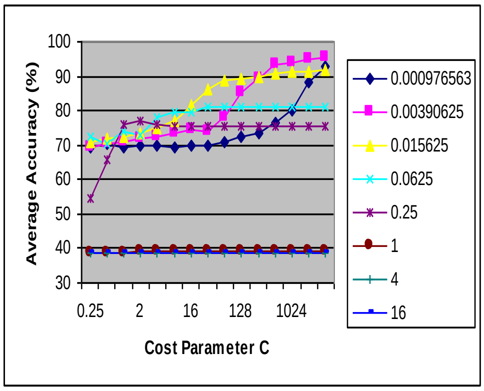
Fig. 2(d) shows the variation of classifier accuracy of iris dataset with γ for different values of C using Hyper Secant kernel function. The variation of classifier accuracy of wine dataset with γ for different values of C using Squared Sinc kernel function is shown in Figure 2(e). Similarly, Figure 2(f) shows the variation of classifier accuracy of glass dataset with γ for different values of C using Laplace kernel function. We have observed from Fig. 2(a)-2(f) that the average cross-validation accuracy of multi-class SVM classifier for a given kernel function depends on the choice of C and γ.

Fig. 3(a)-3(b) shows the variation of classifier accuracy of wine dataset with γ for different kernel functions for C = 4096 and 0.25 respectively. Similarly, Fig. 3(c)-3(d) shows the variation of classifier accuracy of wine data with C for all kernel function for γ = 0.0078 and 0.5 respectively. It can be seen that for certain values of C, variation in certain range of γ does not affect the cross-validation accuracy much, whereas large variation was observed in few cases. We found similar variation in results by varying C and keeping γ being constant.

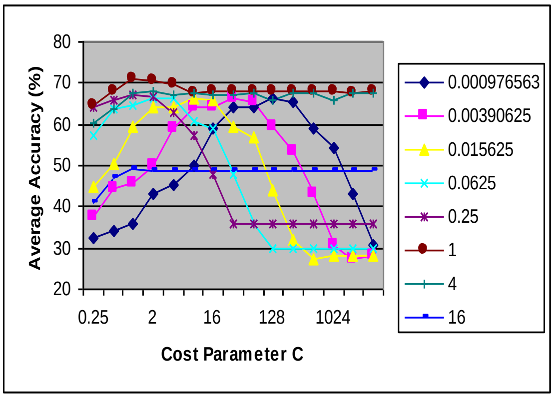
ISSN: 2454-4248
118 – 125



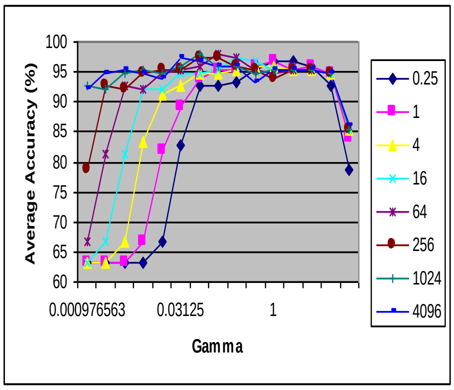
(a)



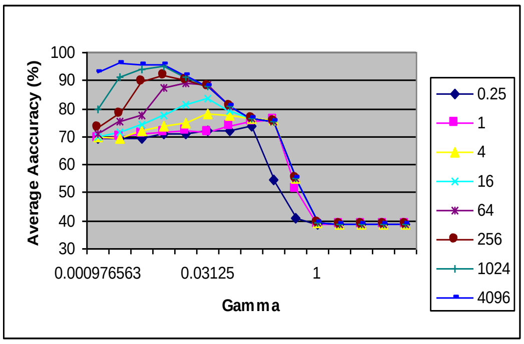
(b)



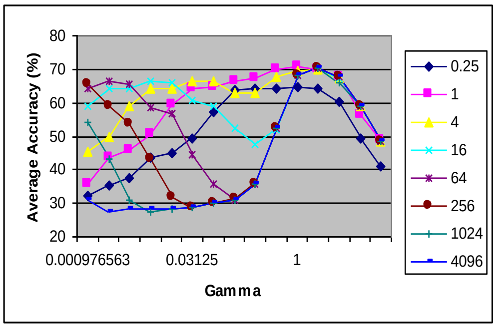
(c)



(d)



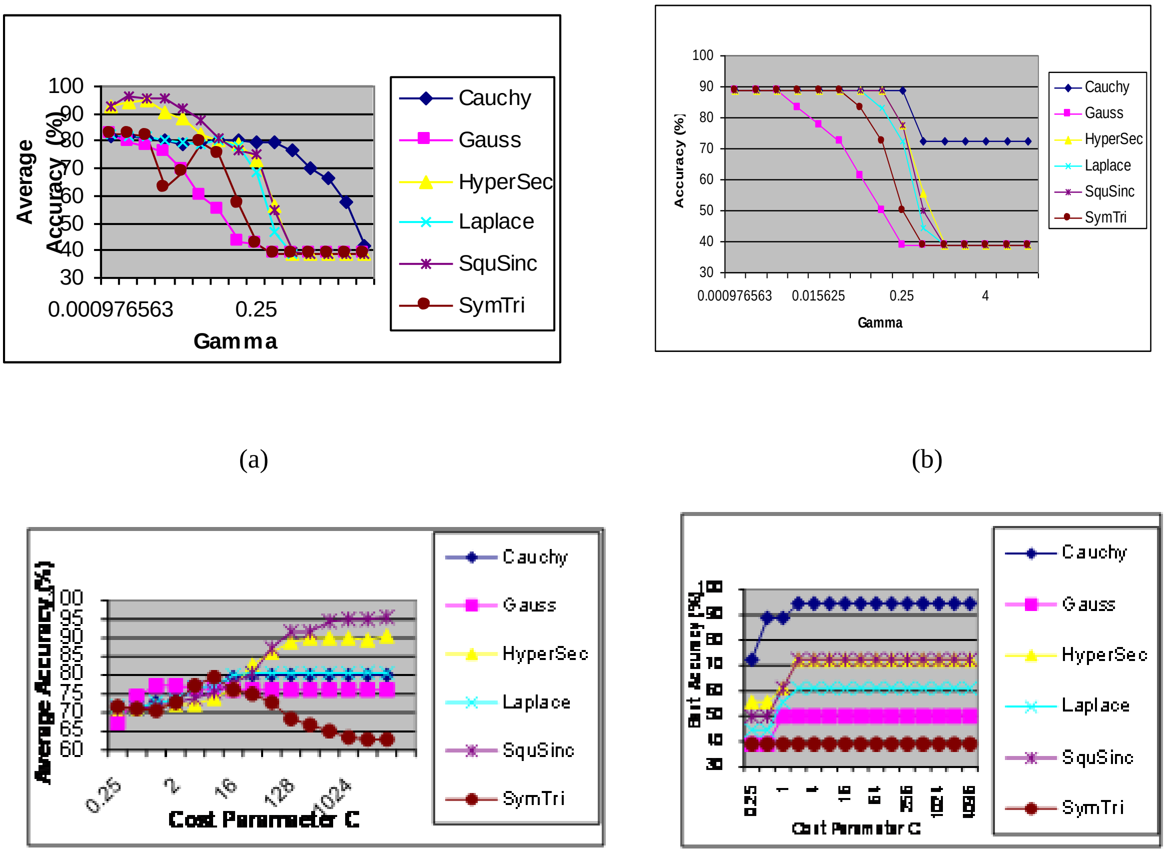
(e)



(f)

(f)Fig.2 Average classifier accuracy for different values of cost parameter and gamma. (a) Iris data using gauss kernel. (b) Wine data using squared sinc kernel. (c) Glass data using laplace kernel. (d) Iris data using hyperbolic secant kernel. (e) Wine data using squared sinc kernel. (f) Glass data using squared sinc kernel.

ISSN: 2454-4248
118 – 125



(a) (b)

(c) (d)

Fig. 3 Classifier accuracy for different kernel functions using wine dataset.

We observed from above figures that accuracy of the classifier for a given dataset is also influenced by the dataset for all possible combinations of C & γ using different kernel functions. The best and average cross-validation accuracy are shown in Table II with their optimal parameters C and γ. For comparison with multi-class SVM, we have also applied decision tree construction algorithm C4.5 [15] on the same datasets for determining the best and average cross-validation accuracy. It can be observed that the best and average cross-validation accuracy using multi-class SVM is same or better than obtained by C4.5 for all datasets. Similarly, Table III compares the accuracy of multi-class SVM classifier with results obtained by C4.5 and available results [9] [20]. The best results in each category are indicated in bold. From Table III, It can be observed that our results are better in each category

Table II

A comparison of classifier accuracy using different kernel functions

Data set	**Kernel →**	**Gauss**	**Cauchy**	**Laplace**	**Hyper Secant**	**Squared Sinc**	**Symmetric Triangle**	**C4.5**
Iris	Best	100 $(2^4,2^{-5})$	100 $(2^8,2^{-10})$	100 $(2^3,2^{-5})$	100 $(2^6,2^{-3})$	100 $(2^2,2^{-1})$	100 $(2^8,2^{-1})$	100
	Average	98 $(2^4,2^{-5})$	96.667 $(2^8,2^{-10})$	96.667 $(2^3,2^{-5})$	98 $(2^6,2^{-3})$	97.333 $(2^2,2^{-1})$	96.667 $(2^8,2^{-10})$	94
Wine	Best	94.444 $(2^5,2^{-10})$	94.444 $(2^9,2^{-10})$	100 $(2^9,2^{-10})$	100 $(2^{12},2^{-9})$	100 $(2^{12},2^{-8})$	100 $(2^{12},2^{-10})$	100

ISSN: 2454-4248
118 – 125

	Average	82.778 $(2^{5},2^{-10})$	81.111 $(2^{9},2^{-10})$	81.667 $(2^{9},2^{-10})$	94.444 $(2^{12},2^{-9})$	96.111 $(2^{12},2^{-9})$	82.222 $(2^{12},2^{-10})$	92.222
Glass	Best	86.364 $(2^{-1},2^{-4})$	77.2727 $(2^{4},2^{1})$	81.818 $(2^{2},2^{1})$	86.364 $(2^{0},2^{0})$	86.364 $(2^{0},2^{0})$	81.818 $(2^{0},2^{-1})$	81.818
	Average	69.091 $(2^{0},2^{1})$	71.818 $(2^{1},2^{2})$	70.909 $(2^{0},2^{0})$	70.909 $(2^{-1},2^{1})$	70.455 $(2^{-1},2^{1})$	69.546 $(2^{0},2^{0})$	71.818

Table III

A comparison of classifier accuracy using different methods for multi-class

Dataset	**Different Methods**	**One against one**	**DAG**	**One against all**	**C & S**			**Ours**
		[9]				[20]	**C4.5**	**One against all**
Iris	Accuracy (C, γ)	97.333 $(2^{12},2^{-9})$	96.667 $(2^{12},2^{-8})$	96.667 $(2^{9},2^{-3})$	97.333 $(2^{10},2^{-7})$	97.333 $(2^{12},2^{-8})$	100	**100** (Gauss,2^{8},2^{-3})
Wine	Accuracy (C, γ)	99.438 $(2^{7},2^{-10})$	98.876 $(2^{6},2^{-9})$	98.876 $(2^{7},2^{-6})$	98.876 $(2^{1},2^{-3})$	98.876 $(2^{0},2^{-2})$	100	**100** (Laplace,2^{9},2^{-10})
Glass	Accuracy (C, γ)	71.495 $(2^{11},2^{-2})$	73.832 $(2^{12},2^{-3})$	71.963 $(2^{11},2^{-2})$	71.963 $(2^{4},2^{1})$	71.028 $(2^{9},2^{-4})$	81.818	**86.364** (HyperSec,2^{0},2^{0})

IV. CONCLUSION

The experimental results of three datasets show that Gaussian kernel is not always the best choice to achieve high generalization of classifier although it often the default choice. We exhibit the dependency of classifier accuracy on the different kernel functions of the multi-class SVM using different dataset. With the choice of kernel function and optimal values of parameters C and γ, it is possible to achieve maximum classification accuracy for all datasets. It will be interesting and practically more useful to determine some method for determining the kernel function and its parameters based on statistical properties of the given data. Then the proposed method in conjunction with multi-class SVM can be tested on application domains such as image processing, text classification, intrusion detection etc. We are also examining the possibility of integrating fuzzy approach in the multi-class SVM classifier.

REFERENCES

[1] C. L. Blake and C. J. Merz. (1998) UCI Repository of Machine Learning Databases. Univ. California, Dept. Inform. Computer Science, Irvine, CA. [Online]. Available: http://www.ics.uci.edu/~mlearn/ML-Repository.html

[2] Y. Chen and J. Z. Wang. "Support vector learning for fuzzy rule-based classification systems," IEEE Transactions on Fuzzy Systems, vol. 11, no. 6, Dec. 2003.

[3] N. Christianni and J. Shawe-Taylor, An introduction to Support Vector machines. Cambridge: Cambridge Univ. Press, 2000

[4] C.J.C. Burges, " A tutorial on support vector machines for pattern recognition," Data Mining Knowledge Discovery, vol. 2, no. 2, pp. 121-167, 1998

[5] C. Corts and V.N. Vapnik, " Support Vector Networks," Machine Learning, Vol. 20, pp. 273-297, 1995

[6] Cover and Hart. "Nearest Neighbor Pattern classification," IEEE transactions on Information Theory, vol. 13, pp. 21-27, 1967

[7] R. Fletcher, Practical Methods of Optimization. John Wiley & Sons, Inc., 2nd edition, 1987.

[8] F. Girosi, M. Jones, and T. Poggio, "Priors, stabilizers and basis functions: From regularization to radial, tensor and additive splines," Technical Report A.I. Memo No. 1430, Massachusetts Institute of Technology, 1993

[9] C.W. Hsu and C. J. Lin, "A comparison of methods for Multi-class Support vector machine," IEEE Transactions on Neural Networks, Vol. 13 (2), 415-425, 2002

[10] Thorsten Joachims. "Text categorization with support vector machines: Learning with many relevant features." In

ISSN: 2454-4248
118 – 125

Proceedings of the European Conference on Machine Learning, Springer, 1998

[11] Thorsten Joachim's, N. Cristianini and J. Shawe Taylor. "Composite Kernels for Hypertext categorization," In proceedings of the International Conference on Machine Learning, 2001.

[12] S. Knerr, L. Personnaz and G. Dreyfus, " Single-layer learning revisited: A stepwise procedure for building and training a neural network. Neurocomputing: algorithms, architectures and applications," Springer.1990

[13] S. Mori, C. Y. Suen and K. Yamamota, "Historical review of OCR research and development," Proceedings of the IEEE, vol. 80, pp. 1029-1058, 1992

[14] S. Mukkamala, G. Janoski and A.H. Sung , "Intrusion Detection using neural networks and support vector machines," Proceedings of IEEE International Joint Conference on Neural Networks, p. 1702-07, 2002

[15] J. Quinlan, "C4.5: program for machine learning," San Mateo: Morgan- Kaufmann, 1993.

[16] R. Rifkin and A. Klautau, "In Defense of One-Vs.-All Classification," Journal of Machine Learning, vol. 5, 101-141, 2004

[17] B. Scholkopf and A. J. Smola, Learning with kernels. MIT Press, Cambridge, MA, 2002

[18] M. Schmidt, "Identifying speaker with support vector networks," In Interface '96 Proceedings, Sydney, 1996

[19] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, Berlin Heidelberg New York, 1995

[20] J. Weston and C. Watkins, "Multi-class Support Vector Machines," presented at the Proc. ESANN99, M. Verleysen, Ed., Brussels, Belgium, 1999.