Volume 4, No. 2, February 2013
Journal of Global Research in Computer Science
ISSN-2229-371X
RESEARCH PAPER
Available Online at www.jgrcs.info

IMPROVING THE CLASSIFICATION ACCURACY USING SUPPORT VECTOR MACHINES (SVMS) WITH NEW KERNEL

Ashraf Afifi and E.A.Zanaty, Said Ghoniemy
IT Department, College of Computers and IT,
Taif City, Taif University, Saudi Arabia,
a.afifi; n.allam@tu.edu.sa

***Abstract**:* In this paper, we introduce a new kernel function called polynomial radial basis function (PRBF) that could improve the classification accuracy of support vector machines (SVMs). The proposed kernel function combines both Gauss (RBF) and Polynomial (POLY) kernels and is stated in general form. It is shown that the proposed kernel converges faster than the Gauss and Polynomial kernels. The accuracy of the proposed algorithm is compared to algorithms based on both Gaussian and polynomial kernels by application to a variety of non-separable data sets with several attributes. We noted that the proposed kernel gives good classification accuracy in nearly all the data sets, especially those of high dimensions.

***Keywords*:** Classification problem, SVMs, kernel functions.

INTRODUCTION

The basic form of support vector machines (SVMs) is to maximize the distance separating the elements of two different classes [1]. When the classes to which the elements belong to are known a priori, the problem is called classification. The set of data used to calculate the boundary limit between the classes is called the training set, while the data set used to test the efficacy of the method, is called validation set. Initially, classification problems were designed to separate two classes, the binary class problem [2, 3].

Despite the maturity of classification, problems remain, especially in choosing the most appropriate kernel of SVMs for a particular application. In recent years, support vector machines (SVMs) have received considerable attention because of their superior performance in pattern recognition and regression [1,4-8]. Choosing different kernel functions will produce different SVMs [7, 10, 11] and may result in different performances. Some work has been done on limiting kernels using prior knowledge, but the best choice of a kernel for a given problem is still an open research issue [12,13]. Comparing SVMs with Gaussian function (GF) to radial basis function (RBF) classifiers can be found in literature [14-17], but these focus on a sub-set of techniques and often only on performance accuracy. Tsang et al. [18] described a way to take advantage of the approximations inherent in kernel classifiers, by using a minimum enclosing ball algorithm as an alternative means of speeding up training. An alternative approach was proposed in [13] to selecting an appropriate kernel is to use invariance transformations.

The drawback here is that they are mostly appropriate only for linear SVM classifiers. The methods of [19-20] can both be applied to pre-image applications with a discrete input space, since they do not require the gradient of the objective function. Generally, in implementations of this method, the time and space complexities are very high because the core of the SVMs is based on approximate minimum enclosing ball algorithms which are computationally expensive. The exact evaluation of intersection kernel SVMs which is logarithmic in time was presented in Maji et al. [21]. They have shown that the method is relatively simple and the classification accuracy is acceptable, but the runtimes are significantly increased compared with the established radial bases function (RBF) and polynomial kernel (POLY) due to large number of SV for each classifier [14, 21]. Zanaty et al. [15-17] combined GF and RBF functions in one kernel called "universal kernel" to take advantage of their respective strengths. The universal kernels constructed the most established kernels such as radial bases, gauss, and polynomial functions by optimizing the parameters using the training data. These kernels satisfied Mercer's condition and converged faster than the existing kernels.

Completely achieving a Support Vector Machine with high accuracy classification therefore requires specifying the high quality kernel function. This paper addresses the problem of data classification using SVMs. We improve the accuracy of SVMs using a new kernel function. We concentrate on non-linearly separable data sets to improve the classification accuracy. The proposed kernel function called polynomial radial basis function (PRBF) that combines both Gauss and Polynomial kernels, is analyzed to prove its advantages over Gaussian and Polynomial kernels. The SVMs modified by the proposed PRBF kernel function is experimented using different data sets.

The rest of this paper is organized as follows: In section 2, the problem formulation is stated. In section 3,m the traditional kernel functions are discussed. A new kernel function is presented and discussed its analysis in section 4. Experimental results are shown in section 5. Finally, section 6 gives our conclusions.

THE PROBLEM FORMULATION

SVMs algorithm [7] has been shown to be one of the most effective machine learning algorithms. It gives very good

results in terms of accuracy when the data are linearly or non-linearly separable. When the data are linearly separable, the SVMs result is a separating hyperplane, which maximizes the margin of separation between classes, measured along a line perpendicular to the hyperplane. If data are not linearly separable, the algorithm works by mapping the data to a higher dimensional featurespace (where the data becomes separable) using an appropriate kernel function and a maximum margin separating hyperplane is found in this space. Thus the weight vector that defines the maximal margin hyperplane is a sufficient statistic for the SVMs algorithm (it contains all the information needed for constructing the separating hyperplane). Since this weight vector can be expressed as a weighted sum of a subset of training instances, called support vectors, it follows that the support vectors and the associated weights also constitute sufficient statistics for learning SVMs from centralized data.

The accuracy problem is usually represented by the proportion of correct classifications. For many data sets, the SVMs may not be able to find any separating hyperplane at all (accuracy equal 0), either because the kernel function is inappropriate for the training data or because the data contains mislabeled examples. The latter problem can be addressed by using a soft margin that accepts some misclassifications of the training examples. A soft margin can be obtained in two different ways. The first is to add a constant factor to the kernel function output whenever the given input vectors are identical. The second is to define a priori an upper bound on the size of the training set weights. In either case, the magnitude of the constant factor to be added to the kernel or to bound the size of the weights controls the number of training points that the system misclassifies. The setting of this parameter depends on the specific data at hand. Completely specifying a support vector machine therefore requires specifying two parameters, the kernel function and the magnitude of the penalty for violating the soft margin. Hence, in order to improve the accuracy of SVMs, we select a suitable kernel function; this is criterion for achieving better results.

Binary Classification Problem:

Binary classification is the simplest one over all the classification tasks. Given a data set D of N samples:(x_1, y_1), (x_2, y_2), ...(x_n,y_n) each sample is composed of a training example $x_i$of length M, with elements $x_i = (x_1, x_2, \ldots, x_m)$, and a target value $y_i \in \{-1,1\}$. The goal is to find a classifier with decision function, $f(x)$, such that $f(x_i) = y_i, \forall (x_i, y_i) \in D$. The performance of such a classifier is measured in terms of the classification error defined in equation (1):

$$error\ (f(x), y) = \begin{cases} 0 & if \quad f(x) = y \\ 1 & otherwise \end{cases} \qquad (1)$$

In order to compute the classification error SVM, we use the Structural Risk Minimization (SRM). The Structural Risk Minimization (SRM) considers the complexity of the learning machine when it searches for α to learn the mapping $x \rightarrow y$. This is done by minimizing the expected risk: $R_{\exp}(\alpha) = \int error(f(x_i, \alpha), y_i) dp(x, y)$, where $p(x,y)$ is a prior probability [22].

Linear Classifiers

There are two cases of linear classifiers. The first where a perfect mapping $x \rightarrow f(x,\alpha)$ can be learned is called the separable case, and the other case where a perfect mapping is unattainable is called the non-separable case [23].

The Separable Case:

Consider the binary classification problem of an arrangement of data points as shown in Fig. (1a) the "square" denotes positive examples with target $y_i = +1$, belonging to the set S_+, and the "round" denotes negative examples with target y_i=-1, belonging to S_-. One mapping that can separate S_+ and S_- is,

$$f(x, y) = sign(w.x + b) \qquad (2)$$

where$w$ is a weight vector and b the offset from origin.

Given such a mapping, the hyperplane, $w.x + b = 0$ (3) defines the decision boundary between S_+ and S_-

The two data sets are said to be linearly separable by the hyperplane if a pair $\{w, b\}$ can be chosen such that the mapping in equation (1) is perfect, this is the case in Fig. (1a). There are numerous values of $\{w, b\}$ that creates separating hyperplanes. The SVMs classifier finds the only hyperplane that maximizes the margin between the two sets (optimal separating hyperplane) this is shown in Fig. (1b).

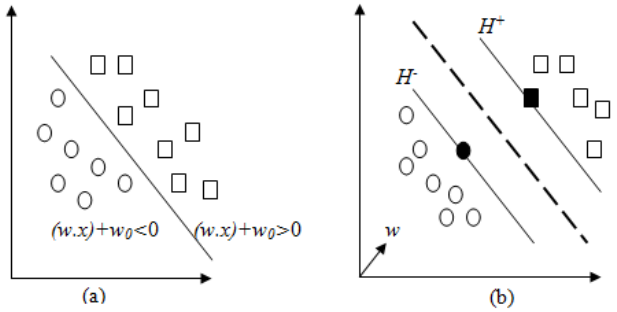

Figure. (1): The hyperplane is separated into two separable sets:a) A separating hyperplane, b) The optimal separating hyperplane.

Consider the problem of separating the set of training vectors belonging to two separate classes,

$$D = \{(x_1, y_1), \ldots\ldots, (x_m, y_m)\}, x \in R^n, y \in \{-1,1\} \qquad (4)$$

with the following decision function,

$$f(x) = sign(w.x + b)$$

If the data is linearly separable then,

$$y_i (w.x + b) > 0 \qquad (5)$$

where w is the normal to the hyperplane, $|b|/\|w\|$ is the perpendicular distance from the hyperplane to the origin, and $\|w\|$ is the Euclidean norm of w. A canonical hyperplane is defined for the support vectors on one side of the separating hyperplane,

$$w.x + b = 1 \qquad (6)$$

for the support vectors on the other side,

$$w.x + b = -1 \qquad (7)$$

and for the separating hyperplane,

$$w.x + b = 0 \qquad (8)$$

if we consider a data point x_1, a support vector on one side of the separating hyperplane and x_2 another support vector on the other side then by substituting into (6) and (7) and subtracting, then,

$$w.x_1 + b - (w.x_2 + b) = w(x_1 - x_2) = 2 \quad (9)$$

for the separating hyperplane the normal vector is,

$$\hat{w} = \frac{w}{\|w\|}, \quad (10)$$

the margin can be defined as half the projection of (x_1-x_2) onto the normal vector giving,

$$2\gamma = \frac{w(x_1 - x_2)}{\|w\|}, \quad (11)$$

from equation (9), we can obtain,

$$2\gamma = \frac{2}{\|w\|},$$

which implies that,

$$\gamma = \frac{1}{\|w\|}. \quad (12)$$

For the linearly separable case, the support vector algorithm looks for the separating hyperplane with largest margin. In order to maximize themargin (γ) the following term is minimized,

$$Min(\frac{1}{2}\|w\|^2), \quad (13)$$

subject to:

$$y_i(w.x_i + b) \geq 1. \qquad \forall i \quad (14)$$

Now, Lagrange multipliers are applied for two reasons.
First, the constraints will be replaced by constraints on the Lagrange multipliers, which will be much easier to handle. The second is that in this first formulation of the problem, the training data will only appear in the form of data products between vectors. This is a crucial property which will allow us to generalize the procedure to the non-linear case. Then Lagrangian is:

$$l = \frac{1}{2}(w.w) - \sum_{i=1}^{m} \alpha_i (y_i (w.x_i + b) - 1),)$$

$$\alpha_i \geq 0, \quad (15)$$

Thus, Lagrange multipliers are, $\alpha_i, \quad i = 1,..., m.$ One for each of the inequality constraints.
If we set ,

$$\frac{\partial l}{\partial w} = 0, \qquad \frac{\partial l}{\partial b} = 0, \quad (16)$$

$$w = \sum_{i=1}^{m} \alpha_i x_i y_i = 0, \quad (17)$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0, \quad (18)$$

Re-substituting (17) and (18) back into (15) obtained which is maximized with respect to α_i,

$$w(\alpha) = \sum_{i=1}^{m} \alpha_i - \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j),$$

subject to,

$$\sum_{i=1}^{m} \alpha_i y_i = 0, \qquad \alpha_i \geq 0.$$

An important detail is that α_i=0 for every $x_i$except the ones that lie on the hyperplanesH^+ and H^-. These points where $\alpha_i \geq 0$ are called support vectors. The number of support vectors in the solution is much less than the number of training examples. This is referred to the sparsity of the solution. When the optimization problem is solved and found the optimal hyperplane, the SVMs can attempt to predict unseen instances [24].

Non-Separable Case:

The SVMs have been restricted to the case where a perfect mapping, $x \rightarrow f(x, \alpha)$, can be learned. Most real-world data sets don't satisfy this condition, so an extension to the above formulation to handle non-separable data is done by creating an objective function that trades of misclassifications against minimizing $||w||^2$. Misclassifications are considered by adding a "slack" variable $\zeta \geq 0$ for each training example, and require that:

$$w.xi - b \geq +1 - \zeta \qquad for\ y_i = +1,$$
$$w.x_i - b \leq -1 + \zeta \qquad for\ y_i = -1.$$

The sum of misclassification is minimized errors as well as minimizing $||w||^2$, $\|w\|^2 + c(\sum_i \zeta_i)\kappa$, where c is a regularization parameter used to control the relation between the slack variables and $||w||^2$, k is an integer with typical values of 1 or 2. This minimization problem is also convex as in the linearly separable case. If we choose k to be 1, it has the advantage that ζ_i's and their Lagrange multipliers disappear from the dual Lagrangian problem.
This objective function of the dual formulation becomes:

$$\text{maximize}\left\{ L_D \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j}^{N} \alpha_i \alpha_j y_i y_j x_i x_j \right\} \quad (19)$$

subject to the constraints,

$$0 \leq \alpha_i \leq c,$$
$$\sum_{i=1}^{N} \alpha_i y_i = 0 \quad (20)$$

When this minimization problem is optimized then:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i,$$

$$b = -\frac{1}{2} w(x_+ + x_-),$$

Where x_+ is the positive example with shortest perpendicular distance from the decision boundary, and x_- is the closest negative example.

The difference between the separable and non-separable is the added constraint in equation (20), now αi's have an upper bound of c.

The support vectors in this solution are not only the training examples that lie on the hyperplane boundary but also the training examples that either falls between the two hyperplanesH_+ and H_- or falls on the wrong side of the decision surface.

Multi-Class Support Vector Machines:

The multi-class problem is defined as the classification problem that has many classes. While SVMs are binary classifiers, i.e. they can classify two classes, we need some techniques to extend these classifiers to handle multiple classes. The goal of such a technique is to map the generalization abilities of the binary classifiers to the multi-class domain. Multi-class SVMs are usually implemented by combining several two class SVMs. In literature, numerous schemes have been proposed to solve this problem, popular methods for doing this are: one-versus-all method using Winner-Takes-All strategy (WTA SVMs), one-versus-one method implemented by Max-Wins Voting (MWV SVMs), DAG SVMs and error-correcting codes[25]. Hsu and Lin [26] compared these methods on a number of data sets and found that MWV SVMs and WTA SVMs give similar generalization performance. Hastie and Tibshirani[27] proposed a good general strategy called pairwise couplingfor combining posterior probabilities provided by individual binary classifiers in order to do multi-class classification and extended it in [28] for speeding up multi-class. Since SVMs do not naturally give out posterior probabilities, they suggested a particular way of generating these probabilities from the binary SVMs outputs and then used these probabilities together with pairwise coupling to do multi-class classification.

Here, we use a multi-class SVM classifier based on one versus one algorithm (the voting strategy) [26, 27], and use the Support Vector Machine toolbox for MATLAB. The classifier is designed to read two input data files, the training data and the test data.

KERNEL FUNCTIONSL

Kernel functions are used to non-linearly map the input data to a high-dimensional space (feature space). The new mapping is then linearly separable [29,30]. The idea of the kernel function is to enable operations to be performed in the input space rather than the potentially high dimension feature space. Hence the inner product does not need to be evaluated in the feature space. The mapping is achieved by replacement of the inner product (x. y)→Φ(x).Φ(*y*) this mapping is defined by the kernel,

$$K(x, y) = \Phi(x).\Phi(y).$$

In order for thedata tobe linearly separable a suitable kernel is chosen. There are many different types of kernels, some of these are listed below.
Consider the linear kernel that just computes the dot product of two vectors, $K(x,z) = x \cdot z = \|x\|\|z\|\cos(u)$, where *u* is the angle between *x* and *z*.

Hence, if the two vectors are orthogonal, their dot product is 0, If they lie in the same direction, their dot product is maximal. By analogy, people tend to think of kernels as similarity functions between input vectors. Not all functions can be used as kernels; feasible kernels must satisfy the following conditions [31],

a. The kernel function must be symmetric.
b. It must satisfy Mercer's theorem [32].

Here are some of the most popular kernels.

Polynomial function*:*

A polynomial mapping is a popular method for non-linear modeling.

$$K(x, x') = (\langle x, x' \rangle)^d .$$
$$K(x, x') = (\langle x, x' \rangle + 1)^d .$$

Gaussian radial basis function*:*

This function has received significant attention, most commonly with a Gaussian of the form,
$K(x , x` )= exp( (-|| x-x`||^2) /(2\sigma^2))$.

Exponential radial basis function*:*

A Radial Basis function of the form,
$K(x , x`)= exp( (-|| x - x`||) / (2\sigma))$,
produces a piecewise linear solution which can be attractive when discontinuities are acceptable.

THE PROPOSED KERNEL FUNCTION

As we mentioned, kernel functions were proposed to handle non-separable data. They are used to map the input data to a high-dimensional space (feature space). So, for a given non-separable data in order to be linearly separable asuitable kernel is chosen. Classical kernels such as Gauss and Polynomial functions, each one performs better with some data sets. Here, we try to formulate a new kernel that could obtain good performance with all data sets and specially high dimension ones (data sets with many attributes).

The following Polynomial function performs good with nearly all data sets, except high dimension ones,

$$POLY = (1+ < x_1, x_2 >)^d ,$$

Where *d* is the polynomial degree. The same performance is obtained with Gauss Radial Basis function of the following form,

$$RBF = \exp(-sum(x_{1i}-x_{2i})^2) / (PD),$$

Where *p* is the kernel parameter, *D* the dimension of the input vector (number of attributes).

We propose a new form of kernel functions which is more complex that could handle high dimension data sets, we denote it as PRBF. This kernel combines both Gauss and Polynomial functions, so it performs better with nearly all data sets,

$$PRBF = (1+ \exp(-sum(x_{1i}-x_{2i})^2) / (PD))^d .$$

The performance of this kernel is shown later by the experimental results.

The proposed kernel function (PRBF) has large convex than the classical Polynomial and Gaussian functions. In addition

this function is decreasing when x decreases and it is continuous. If we calculate the limit of (PRBF/RBF) we will get,

let $T = \sum (x_{1i} - x_{2i})^2$ and $V = PD$,

then,

$$PRBF = \left(1 + \frac{e^{-T}}{V}\right)^d \quad \text{and} \quad RBF = \frac{e^{-T}}{V},$$

hence,

$$\lim_{x_{1i}, x_{2i} \to \infty} \left(\frac{\left(1 + \frac{e^{-T}}{V}\right)^d}{\frac{e^{-T}}{V}} \right) = \lim_{x_{1i}, x_{2i} \to \infty} \left(\frac{\left(1 + \frac{1}{e^{T}}\right)^d}{\frac{1}{e^{T}}} \right) = \infty,$$

Therefore, the proposed function converges faster than the Gauss function. We normalized the data sets to be within the interval [-1,1], because the Polynomial function will diverge in large intervals, and the proposed function has faster convergence within this interval. The following Fig. (2), Fig. (3) and Fig. (4) show the shape of Polynomial (POLY), Gauss (RBF) and the proposed (PRBF) kernels, respectively, within the interval [-1,1] and with two-dimension vectors x1,x2.

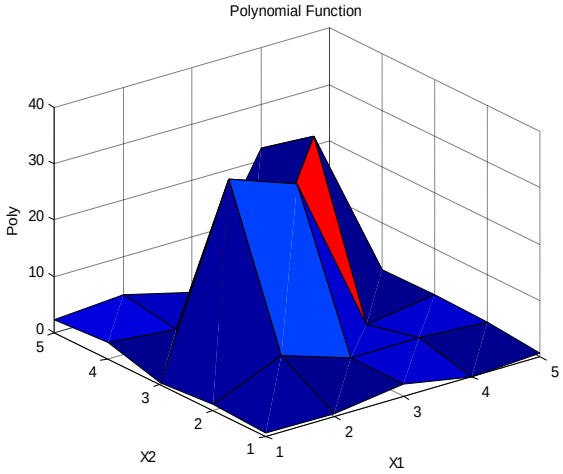

Figure. (2): The Polynomial kernel

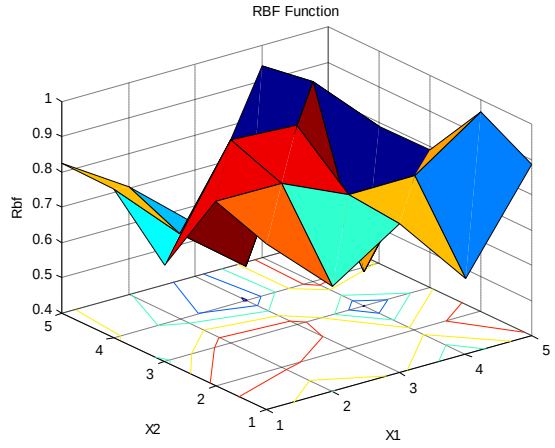

Figure. (3): The Gauss kernel

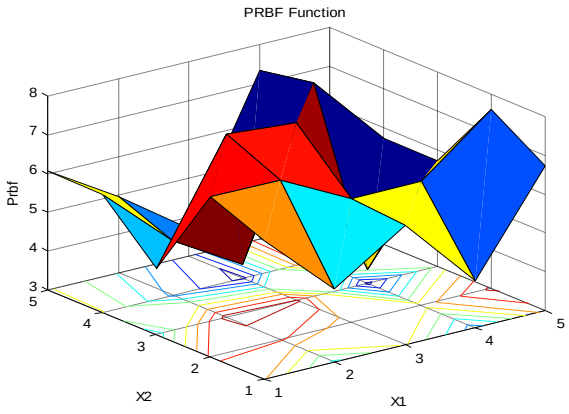

Figure.(4): The proposed kernel

EXPERIMENTAL RESULTS

Data Sets:

In order to evaluate the performance of the proposed kernel with SVMs, we carried out some experiments with different data sets. Table (1) shows the description of these data sets (see Appendix) and for more details can be seen in [33,34]. We can divide these data sets according to the training set size into two types, large data sets (1→4) and small ones (5→8). Fig.(5) shows some examples for the input data and the output.

Table (1): Data Sets

N0.	**Data set**	Classes	Attributes	Training	Test
1	Letter	26	16	15000	5000
2	Pendigits	9	16	7435	3448
3	Waveform	3	21	4700	300
4	Satimage	6	36	4435	2000
5	DNA	3	180	2686	500
6	Segment	7	18	1810	500
7	ABE	3	16	1763	560
8	Zoo	7	17	70	31

	x_1	x_2	…	x_m	y
X_1	0.01187666	1.01381835	…	0.21932496-	1
X_2	0.79640450	.61823882	…	2.09936881-	2
X_3	0.04157263	1011182652.	…	10.4496266	3
…					
X_n	...	…	…	…	y_c

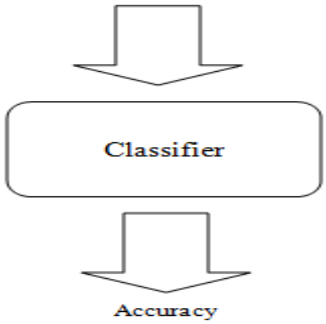

Figure. (5): Example for input/output to the classifier.

Comparative results:

We design a multi-class SVM classifier based on one versus one algorithm (the voting strategy), and we use the Support Vector Machine toolbox for MATLAB. We design the classifier to read two input data files (as shown in Fig.(5), the training data and the test data. Each file is organized as

records, each of which consists of a vector of attributes $X=[x_1, x_2, ..., x_M]$ followed by the target $Y \in \{y_1, y_2, ..., y_C\}$ where M is the number of attributes and C is the number of classes. It constructs $C(C\text{-}1)/2$ binary classifiers, and uses the training data to find the optimum separating hyperplane. Finally, we use the test data to compute the accuracy of our classifier from the equation, $Acc = (n / N) *100$, where n is the number of correct classified examples and N is the total number of the test examples. We compare the results of Polynomial, Gauss and the proposed kernels with the classifier as in Table (2).

Table (2): SVMs classification accuracy.

Data Set	RBF(σ =3)	POLY(d=4)	PRBF(P,d=3)
1	93.9	93.9	93.9
2	82	82.52	83.67
3	95	98.67	99
4	92.95	96.7	96.1
5	94.86	89.46	98.8
6	97	92.12	99
7	100	99.82	99.82
8	87.09	87.09	90.32

Result analysis:

From Table (2) it is obvious that Gauss Radial Basis function with σ =3 accomplishes better accuracy with the small data sets (5→8) than the Polynomial function, and the Polynomial kernel with d=4 gives better results in the large sets (1→4). Whereas our proposed kernel, PRBF, accomplishes the best accuracy in nearly all the data sets, and especially in the largest number of attributes data set number (5), because the proposed function is more complex and combines the performance of both its parents, Gauss and Polynomial functions.

Table (3) presents the mean accuracy we obtained from all kernels; it is obvious that the new proposed kernel obtains the best mean accuracy compared to the classical Gauss and Polynomial function. Fig. (6) Summarizes the comparison of the performance of the SVMs with different kernels (POLY, RBF & PRBF), it is clear that the proposed kernel (PRBF) achieves the highest accuracy.

Table (3): SVM mean accuracy.

Kernel	Mean accuracy
RBF	92.85
POLY	92.53
PRBF	95.08

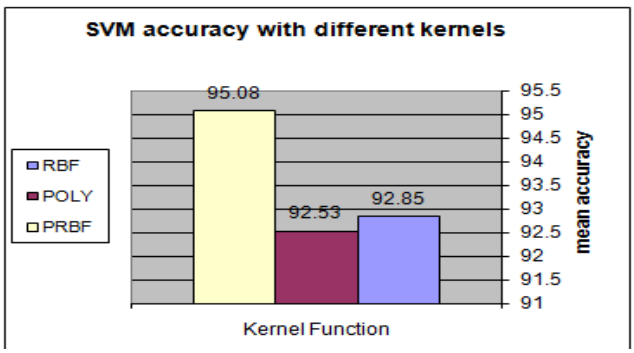

Figure.(6): SVM mean accuracy.

CONCLUSION

In this paper, SVMs have been improved to solve the classification problems by mapping the training data into a feature space by the aid of new kernel functions and then separating the data using a large margin hyperplane.

We have tested the proposed kernel function with different sizes of data sets and different attributes. It is obvious from experimental results that RBF gives better accuracy with the small data sets than the Polynomial function. However the Polynomial kernel gives better results in the large data sets. Whereas our proposed kernel PRBF obtains the best accuracy in nearly all the data sets and especially in the largest number of attributes data set, because the proposed function combines the performance of both its parents, Gauss and Polynomial functions.

Experimental results have been reported to illustrate the validity and effectiveness of the proposed kernel. The experimental results show that the proposed kernel function obtains the best accuracy in nearly all the data sets especially in the largest number of attributes data set. Thus the proposed kernel functions can be considered as a good alternative to the Gaussian and polynomial kernel functions for some specific datasets.

REFERENCES

[1]. Vapnik V. N., "The nature of statistical learning theory", Springer-Verlag, New York, NY, USA, 1995.

[2]. Kim H., Pang S., Je H., Kim D., Bang S.Y., "Constructing support vector machine ensemble", Pattern Recognition", vol.36, no.12, pp.2757–2767, 2003.

[3]. Du P., Peng J., TerlakyT.. "Self-adaptive support vector machines", modeling and experiments Computational Management Science, vol. 6, no.1, pp. 41–51, 2009.

[4]. Boser B. E., Guyon I. M., Vapnik V. N., "A training algorithm for optimal margin classifiers", Proc. Fifth Ann. Workshop Computing Learning Theory, pp. 144-152, 1995.

[5]. Vapnik V.N., "An overview of statistical learning theory", IEEE Trans. Neural networks, vol. 10, no. 5, pp. 988-999,1999.

[6]. Cortes C., Vapnik V.N., "Support-vector networks", Machine Learning, vol. 20, pp. 273- 297,1995.

[7]. Burges, C., "A tutorial on support vector machines for pattern recognition", Data Mining and Knowledge Discovery 2 (2), pp. 67-121, 1998.

[8]. Vapnik V.N., Golowich S., and SmolaA.,"Support vector method for function approximation, regression estimation and signal processing", Advances in Neural Information processing Systems, vol. 9, Cambridge, Mass.: MIT Press, 1997.

[9]. Mangasarian O.L., Musicant D.R., "Successive over relaxation for support vector machines", IEEE Trans. Neural Networks, vol. 10, no. 5, pp. 1032-1037,1999.

[10]. Aronszajn N. , "Theory of Reproducing Kernels", Trans. Am. Math. Soc.,vol. 68, pp. 337-404, 1950.

[11]. Shawe-Taylor J., Bartlett P.L., Willianmson R.C., Anthony M., "Structural risk minimization over data-dependent hierarchies", IEEE Trans. Information Theory, vol. 44, no. 5, pp. 1926-1940, 1998.

[12]. Williamson R. C., Smola A., Schölkopf B., "Entropy numbers, operators and support vector kernels", MA: MIT Press, Cambridge, pp. 44-127, 1999.

[13]. Chapelle O. and Schölkopf B., "Incorporating invariances in non-linear support vector machines", In T. G. Dietterich, S. Becker, and Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems, Cambridge, MA: MIT Press, vol. 14, pp. 594-609, 2002.

[14]. Hastie T., Hsu C-W., Lin C-J, "A comparison of methods for multi-class support vector Machines", IEEE Transactions on Neural Networks, no. 13, pp.415–425, 2005.

[15]. Zanaty , E.A. ,and Sultan Aljahdali, "Improving the accuracy of support vector machines", Proceedings of 23rd International conference on computers and their application April, pp. 75-83, Cancun, Mexico, 2008.

[16]. Zanaty E.A, Sultan Aljahdali, R.J. Cripps, "Accurate support vector machines for data classification", Int. J. Rapid Manufacturing, vol. 1, no. 2, pp. 114-127, 2009.

[17]. Zanaty E.A, Ashraf Afifi,"Support vector machines (SVMs) with universal kernels ", in International journal of Artificial Intelligence, vol. 25, pp.575-589, 2011.

[18]. Tsang I. W., Kwok J. T., Cheung P.-M., "Core vector machines: Fast SVMs training on very large data sets", Journal of Machine Learning Research, vol. 6, pp. 271-363, 2005.

[19]. Kwok J. T., Tsang I. W, "The pre-image problem in kernel methods", IEEE Trans.On Neural Networks, no. 15, pp. 1517–1525, 2004.

[20]. Bakir G. H. , Weston J., Schlkopf B., "Learning to find pre-images", Advances in Neural Information Processing Systems, vol. 16, pp. 449–456, 2004.

[21]. Maji S., Berg A.C., Malik J., "Classification using intersection kernel support vector machines is efficient", IEEE Conference on Computer Vision and Pattern Recognition, June pp.1–8, 2008.

[22]. Boser, B., I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers", In D. Haussler (Ed.), Annual ACM Workshop on COLT, Pittsburgh, PA,ACM Press, vol. 144, no.52, 1992.

[23]. Osuna, E., R. Freund, and F. Girosi, "An improved training algorithm for support vector machines", In J. Principe, L. Gile, N. Morgan, and E. Wilson (Eds.), Neural Networks for Signal Processing VII: Proceedings of IEEE Workshop, New York, NY, pp. 191-276, 1997.

[24]. Burges, C. and B. Schölkopf, "Improving the accuracy and speed of support vector machines", Advances in Neural Information Processing Systems, vol. 9, no.375, 1997.

[25]. Rifin R.,Klautau A., "In defense of one vs. all classification", Journal of Machine Learning Research, vol.5, pp.101-141, 2004.

[26]. Hsu C.-W., Lin C.-J., "A comparison of methods for multi-class support vector Machines", IEEE Transactions on Neural Networks, vol.13, pp. 415-425, 2002.

[27]. Hastie T. &Tibshirani R., "Classification by pairwise coupling", Annals of Statistical, vol.26, no.2, pp. 451-471, 1998.

[28]. Hansheng Lei,VenuGovindaraju, "Speeding up multi-class SVM evaluation by PCA and feature selection", Center for Unified Biometrics and Sensors (CUBS), 2005.

[29]. Gunnar Ratsch, "A brief introduction into machine learning", Friedrich Miescher Laboratory of the Max Planck Society, Germany, 2005.

[30]. Scholkopf B., Smola A.J., "Learning with kernels, support vector machine, Regularization, Optimization and Beyond", Cambridge, MA.MIT Press, 2002.

[31]. Muller K.-R., S. Mika G., Ratsch, Tsuda K., Scholkopf B., "An introduction to kernel-based learning algorithms", IEEE Transactions on Neural Networks, vol.12, no.2, pp. 181–201, 2001.

[32]. Boudat G., Anour F., "Kernel-based methods and function approximation", International Joint Conference on Neural Networks (IJCNN), Washington, IEEE Press, vol.1, 2001.

[33]. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[34]. http://www.liacc.up.pt/ML/old/statlog/datasets.html1.

