Accurate support vector machines for data classification

E.A. Zanaty* and Sultan Hamadi Aljahdali

College of Computer Science,
Taif University,
Hawya, P.O. Box 888, Taif, Saudi Arabia
E-mail: zanaty22@yahoo.com
E-mail: n.allam@tu.edu.sa
E-mail: aljahdali@tu.edu.sa
*Corresponding author

R.J. Cripps

Geometric Modeling Group,
School Mechanical Engineering,
University of Birmingham,
Edgbaston, Birmingham, B15 2TT, UK
E-mail: r.cripps@bham.ac.uk

Abstract: In this paper, a new kernel function is introduced that improves the classification accuracy of support vector machines (SVMs) for both linear and non-linear data sets. The proposed kernel function, called Gauss radial basis polynomial function (RBPF) combine both Gauss radial basis function (RBF) and polynomial (POLY) kernels. It is shown that the proposed kernel converges faster than the RBF and POLY 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. It is shown that the proposed kernel gives good classification accuracy in nearly all the data sets, especially those of high dimensions.

Keywords: kernel functions; support vector machines; SVMs; classification problem.

Reference to this paper should be made as follows: Zanaty, E.A., Aljahdali, S.H. and Cripps, R.J. (2009) 'Accurate support vector machines for data classification', *Int. J. Rapid Manufacturing*, Vol. 1, No. 2, pp.114–127.

Biographical notes: E.A. Zanaty is an Associate Professor and the Head of Computer Science Department, Faculty of Computer and Information Systems, Taif University, Saudi Arabia. He received his MSc in Computer Science in 1997 from South Valley University, Egypt. He completed his PhD studies at TU-Chemnitz, Germany, during the period 2000–2003. His research interests are reverse engineering, data reduction, data segmentation and classification, and image reconstruction.

Sultan Hamadi Aljahdali received his BS from Winona State University, Winona, Minnesota in 1992, MS with honour from Minnesota State University, Mankato, Minnesota in 1996 and PhD in Information Technology from George Mason University, Fairfax, Virginia, USA in 2003. He is the Dean of the

College of Computers and Information Systems at Taif University. His research interest includes software testing, developing software reliability models, soft computing for software engineering, computer security, reverse engineering, and medical imaging; furthermore he is a member of ACM, IEEE, and ISCA.

R.J. Cripps was appointed a Lecturer in Mechanical Engineering at the University of Birmingham in 1989 and has played a leading role in the CAE activities of GMG, with overall responsibility and strong research interests in computing and computer graphics. In 2005 he became the Head of Geometric Modelling Group and was appointed a Senior Lecturer in 2009. He has played a key role in the development of the point-based approach to surface modelling.

1 Introduction

Support vector machines (SVMs) were introduced by Boser et al. (1992) to solve classification problems that commonly arise in machine learning operations. Although, it is a relatively new field of research, there exists several classification learning algorithms. However, there still exist issues regarding the accuracy of data classification using SVMs.

One issue is identifying an appropriate kernel for the given data. Most algorithms rely on a priori knowledge to select the correct kernel. This is then followed by optimisation of the kernel parameters via machine learning or trial-and-error. Whilst rules-of-thumb exist for choosing appropriate kernel functions and parameters, this limits the usefulness of SVMs to expert users, especially since different functions and parameters can have widely varying performance. In an attempt to develop sound reasoning to identify a suitable kernel, Williamson et al. (1999) proposed using entropy numbers associated with mapping operators for Mercer kernels to aid identification of an appropriate kernel function. They derived new bounds for the generalisation error of feature space machines, such as SVMs and related regularisation networks, by obtaining new bounds on their covering numbers. The proofs are based on a viewpoint that is apparently novel in the field of statistical learning theory. The hypothesis class is described in terms of a linear operator mapping from a possibly infinite dimensional unit ball in feature space into a finite dimensional space. The covering numbers of the class are then determined via the entropy numbers of the operator. These numbers, which characterise the degree of compactness of the operator, can be bounded in terms of the eigenvalues of an integral operator induced by the kernel function used by the machine (Williamson et al., 1999; Chapelle and Schölkopf, 2002). An alternative approach to selecting an appropriate kernel is to use invariance transformations (Chapelle and Schölkopf, 2002). The drawback here is that they are mostly appropriate only for linear SVM classifiers. An analytical method that uses a kernel principal component analysis map incorporating invariance transformations (Lei and Govindaraju, 2005) can be applied to non-linear SVM classifiers. However, the SVM itself also needs improvement in both training and testing (evaluation).

Tsang et al. (2005) discussed 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. Training time had previously been reduced mostly by modifying the training set in some way. Their core vector machine converged in linear

time with space requirements independent of the number of data points (Tsang et al., 2005). 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.

Maji et al. (2008) presented a technique for the exact evaluation of intersection kernel SVMs which is logarithmic in time. 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 RBF and POLY kernels due to large number of SV for each classifier.

Completely achieving a SVM with high accuracy classification therefore, requires specifying high quality kernel function. This paper addresses the problem of data classification using SVMs and considers as an extension of Zanaty and Aljahdali (2008) and Nashaat (2008) for constructing an SVM with high classification accuracy. We improve the accuracy of SVMs by introducing a new kernel function that performs better than existing kernel functions (Zanaty and Aljahdali, 2008). The proposed kernel function combines both polynomial (POLY) and RBF. The improved classification accuracy is demonstrated empirically by application to a variety of non-linear datasets.

The rest of this paper is organised as follows: In Section 2, the problem formulation is stated. In Section 3, kernel functions are discussed, where the new kernel functions is introduced and analysed. A comparison between the proposed and existing kernels is given in Section 4. Finally, Section 5 summarises the properties and advantages of the proposed kernel functions and conclusions are given.

2 Problem formulation

The accuracy problem is usually represented by the proportion of correct classifications. For many data sets, the SVMs may have zero accuracy, either because the kernel function is inappropriate for the training data or because the data contains mislabelled examples. The latter problem can be addressed by using a soft margin, or tolerance, 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 bind 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. Completely specifying an SVM therefore, requires specifying two parameters, the kernel function and the magnitude of the penalty for violating the soft margin. The focus here is to specify the kernel function and it is assumed that the soft margin is the most appropriate for a given data set.

2.1 Support vector machine (SVM)

The basic form of a SVM classifier can be expressed as:

$$g(x) = w.\varphi(x) + b \tag{1}$$

where $x \in R^n$ is the input vector, w is a normal vector of a separating hyperplane in the feature space produced from the mapping of a function $\varphi(x): R^n \to R^{n'}$ ($\varphi(x)$ can be linear or non-linear, n can be finite or infinite), and b is a bias scalar. Since SVMs were originally designed for two-class classification, the sign of $g(x)$ associates vector x to class 1 or class –1.

Given a set of training samples x, $x_i \in R^n$, i = 1, 2, .., n and corresponding labels $y_i \in \{-1, +1\}$, the separating hyperplane (described by w) is determined by minimising the structure risk instead of the empirical error (Boser et al., 1992). Minimising the structure risk is equivalent to seeking the optimal margin between two classes. The width of the margin is $\frac{2}{w.w} = \frac{2}{\| w \|^2}$. For SVM classification training involves the minimisation of the error function (Boser et al., 1992):

$$\min \frac{1}{2} w.w + C \sum_{i=1}^{N} \xi_i \tag{2}$$

subject to $y_i (w.\varphi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i,$ where the scalar parameter C is the trade-off.

The solution to (2) is reduced to solving a quadratic programming (QP) optimisation problem:

$$\text{Max } \alpha^T \alpha - \frac{1}{2} \alpha^T H \alpha \tag{3}$$

subject to $0 \leq \alpha_i \leq C, \forall i, \sum_{i=1}^{N} y_i \alpha_i = 0,$ where $\alpha = [\alpha_i, .., \alpha_N]^T$, and H is a $N \times N$ matrix, called the kernel matrix, with each element: $H_{i,j} = y_i y_j \phi(x_i).\phi(x_j)$.

Solving the QP problem yields:

$$\begin{aligned} w &= \sum_{i=1}^{N} \alpha_i y_i \varphi(x_i), \\ b &= \sum_{j=1}^{N} \alpha_i y_j \varphi(x_i).\varphi(x_j) + y_i, \forall i \end{aligned} \tag{4}$$

Each training sample x_i is associated with a Lagrange coefficient α_i. Those samples whose coefficient α_i is nonzero are called support vectors (SV). Only a small portion of training samples become SVs (typically of the order of 3%). Substituting (4) into (1) gives the formal expression of the SVM classifier:

$$\begin{aligned} g(x) &= \sum \alpha_i y_i \varphi(x_i).\varphi(x) + b \\ &= \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b, \end{aligned}$$

where K is a kernel function: $K(x_i, x_j) = \varphi(x_i).\varphi(x_j)$. The most commonly-used kernel functions are:

1 linear i.e., $\varphi(x_i) = x_i$, thus, $K(x_i, x_j) = x_i.x_j = x_i^T x_j$

2 polynomial i.e., $K(x_i, x_j) = (x_i.x_j + c)^d$, where c and d are prescribed constants.

3 Gaussian radial basis (RBF) i.e., $K(x_i, x_j) = e^{(\frac{\|x_i - x_j\|^2}{2\sigma^2})}$.

If the chosen kernel is linear, then the SVM is called linear, otherwise, it is a non-linear SVM. Since the training problem has been solved by using sequential minimum optimisation using existing kernels (Lei and Govindaraju, 2005), the focus of efforts has been transferred to the evaluation of SVMs (Lei and Govindaraju, 2005; Tsang et al., 2005; Maji et al., 2008) in attempts to improve evaluation speed and the challenging issue of accuracy.

2.2 Multi-class SVMs

The multi-class problem is defined as the classification problem that has many classes or attributes. Existing SVMs (Lei and Govindaraju, 2005) are binary classifiers, i.e., they can classify two classes. To be able to cope with multiple (> 2) classes, the existing classifiers have to be extended. The goal is to map the generalisation abilities of the binary classifiers to the multi-class domain. Multi-class SVMs are usually implemented by combining several two class SVMs. In the literature, numerous schemes have been proposed to solve this problem including: one-versus-all methods using the winner-takes-all strategy (WTA SVMs); one-versus-one methods implemented by max-wins voting (MWV SVMs) and error-correcting codes (Rifin and Klautau, 2004). Hsu and Lin (2002) compared these methods on a number of data sets and found that MWV- and WTA-based SVMs give similar performances. Hastie and Tibshirani (1998) proposed a good general strategy called pair wise coupling for combining posterior probabilities provided by individual binary classifiers. This was extended in Lei and Govindaraju (2005) to improve the classification speed. Since SVMs do not naturally give out posterior probabilities, they suggested a practical method of generating these probabilities from the binary SVMs outputs. These probabilities together with pair wise coupling are then used to carry out the multi-class classification. In the implementations of the methods given in Hastie and Tibshirani (1998) and Lei and Govindaraju (2005), the time complexity is high, much time is consumed in solving the optimisation problem directly on the core set.

The proposed method adopts a multi-class SVM classifier based on the voting strategy (MWV) (Rifin and Klautau, 2004; Sonnenberg et al., 2006; Cuturi, 2007; Haasdonk, 2005) due to computational simplicity, and can be implemented using the SVM toolbox within MATLAB (Maji et al., 2008). The classifier is designed to read two input data files, the training data and the test data (for more details see Maji et al., 2008).

3 Kernel functions

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 (Maji et al., 2008; Ratsch, 2005; http://www.ics.uci.edu/~mlearn/MLRepository.html.). 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. For example, the inner product does not need to be evaluated in the feature space. The mapping is achieved by replacing the inner product $(x.y) \rightarrow \Phi(x).\Phi(y)$. This mapping is defined by the kernel:

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

For given non-separable data, in order to be linearly separable, a suitable kernel has to be chosen. Classical kernels, such as Gauss and POLY functions, can be used to transfer non-separable data to separable, but their performance in terms of accuracy is dependent on the given data sets.

The following POLY function performs well (Maji et al., 2008) with nearly all data sets, except high dimension ones:

$$POLY = (1 + \langle x_1, x_2 \rangle)^d$$

where d is the polynomial degree. The same performance (Maji et al., 2008) is obtained with the Gauss RBF of the following form:

$$RBF = \frac{e^{-\sigma}}{pD} = \frac{e^{-\sigma}}{V}$$

where $V = pD$, and $\sigma = \sum_{i=0}^{m} (x_{1i} - x_{2i})^2$, p is the kernel parameter and D is the dimension of the input vector (number of attributes).

3.1 Proposed kernel function

A new class of kernel function is proposed that can handle high dimension data sets and is computationally efficient when handling non-separable data with multi attributes. It has been shown that POLY has a better classification accuracy than RBF (Lei and Govindaraju, 2005) whilst RBF is faster than POLY (Tsang et al., 2005). The proposed method simply combines both Gauss RBF and POLY kernels to take advantage of their respective strengths. The proposed kernel is given by:

$$PRBF = \left(\frac{(1 + e^{\theta})}{V} \right)^d$$

where $\theta = \sum_{i=0}^{m} | x_{1i} - x_{2i} |$.

3.2 Proposed kernel function analysis

The geometric characteristics of the three kernel functions are illustrated in Figures 1–3. By comparing the geometric shapes of these functions, it can be seen that the proposed kernel function (RBPF) (Figure 3) has a large convex region compared to the classical POLY (Figure 1) and Gaussian (Figure 2) functions. In addition, the proposed function (RBPF) is continuous and decreasing when sampling x increases, an important property for improved accuracy and evaluation speed of any SVMs algorithm.

The convergence of the proposed function can be determined by evaluating the limit of RBPF at $\theta = 0$ and infinity:

$$\lim_{\sigma\to 0}\left(\frac{(1+e^{-\theta})^d}{V}\right)\Rightarrow\left(\frac{2}{V}\right)^d$$

$$\lim_{\sigma\to \infty}\left(\frac{(1+e^{-\theta})^d}{V}\right)\Rightarrow\frac{1}{V^d}.$$

This proves that RBPF is convergent and does not depend on the size of the dataset. Moreover, the behaviours of the corresponding POLY and Gaussian (RBF) can be estimated also by considering the limit of each kernel function individually.

In the case of POLY:

$$\lim_{\sigma\to 0}(1+(x_{1i}\cdot x_{2i})^d)\Rightarrow 1$$

$$\lim_{\sigma\to \infty}\left(1+(x_{1i}\cdot x_{2i})^d\right)\Rightarrow \infty$$

In the case of RBF:

$$\lim_{\sigma\to 0}\left(\frac{e^{-\theta}}{V}\right)\Rightarrow\frac{1}{V}$$

$$\lim_{\sigma\to \infty}\left(\frac{e^{-\theta}}{V}\right)\Rightarrow 0$$

RBPF is bounded and convergent for any dataset. This is not true in the case of the POLY kernel function. Further, normalising the data sets to be within the interval [–1, 1] avoids the possibility that the POLY function will diverge. Also, accurate results can be achieved in RBPF by estimating a suitable choice for the degree d using the training data; this opportunity does not exist in the RBF kernel.

As an independent test, the geometric shape of the three kernel functions, POLY, RBF, and RBPF constructed on the same datasets x_1, x_2 are shown in Figures 1, 2 and 3 respectively. The shape of POLY, Gauss (RBF) and the proposed (RBPF) kernels, respectively, is estimated within the interval [–1, 1] using two-dimensional vectors; $x_1 = (x_{11}, x_{12})$, $x_2 = (x_{21}, x_{22})$, and the degree of the POLY, d, is fixed at three. POLY does not coverage at all, RBF is convergent and bounded at 0.95, and RBPF is convergent and bounded at 0.85. This interval also ensures that the proposed function has faster convergence and gives a large convex region. In the next section, the improved

performance of this kernel is demonstrated by application to a variety of data sets, including non-separable and sets with many attributes.

Figure 1 The polynomial kernel (see online version for colours)

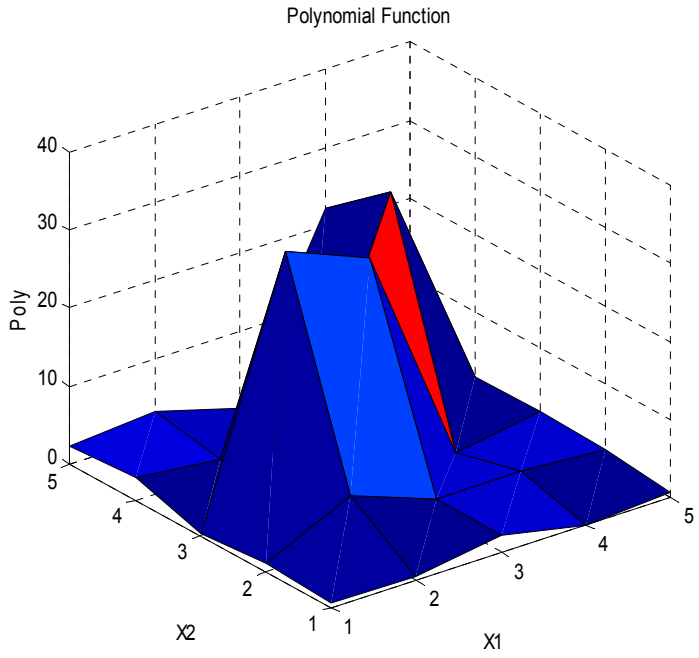


Figure 2 The Gauss kernel (see online version for colours)

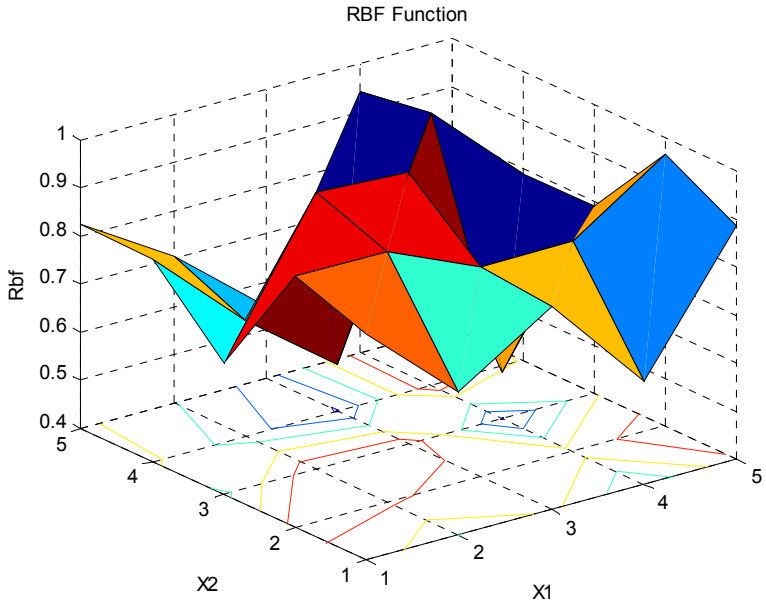
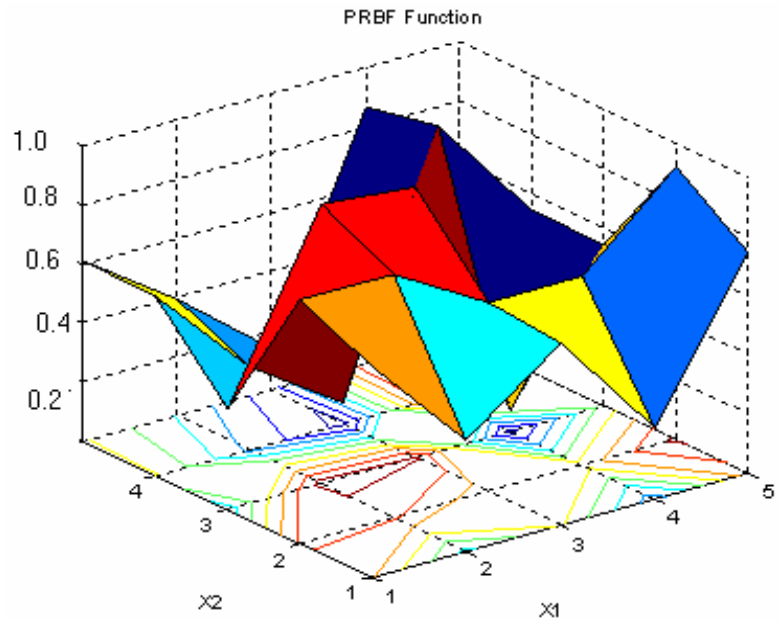


Figure 3 The proposed kernel (see online version for colours)



4 Comparison between the proposed and traditional kernels

4.1 Data sets

The performance of the proposed kernel with SVMs, in terms of classification accuracy, is evaluated by application to a variety of data sets. Table 1 shows the description of these data sets originally used in http://www.ics.uci.edu/~mlearn/MLRepository.html and http://www.liacc.up.pt/ML/old/statlog/datasets.html1, and can be characterised according to the training set size leading to large (sets 1 → 4) and small (sets 5 → 8). A multi-class SVM classifier was designed based on the one versus one algorithm and implemented using the SVM toolbox within MATLAB, the main features of which are:

1 all parts are written in plain MATLAB guaranteeing ease of modification

2 special kinds of kernels that require much computation such as the Fisher kernel, which is based on a model of the data, can easily be incorporated

3 extension to multi-class problems via error correcting output codes is included

4 unlike many other SVM toolboxes, MATLAB SVM toolbox can handle SVMs with 1- or 2-norm of the stack variable

5 for both 1- and 2-norm, a decomposition algorithm is implemented for the training routine, together with efficient working set selection strategies.

Table 1 Data sets

No.	*Data set*	*Classes*	*Attributes*	*Training*	*Test*
1	Letter	26	16	15,000	5,000
2	pendigits	9	16	7,435	3,448
3	waveform	3	21	4,700	300
4	Satimage	6	36	4,435	2,000
5	DNA	3	180	2,686	500
6	Segment	7	18	1,810	500
7	ABE	3	16	1,763	560
8	Zoo	7	17	70	31

4.2 Experiments

In order to determine the performance characteristics of an SVM based on the proposed kernel, a testbed was constructed using an MWV SVM classifier (Rifin and Klautau, 2004), and the SVM toolbox within MATLAB. Since a support vector is expressed by a quadratic optimisation problem, the solution is globally optimal. However, because of the use of nonlinear kernels, the dual optimisation problem has to be solved whose number of variables is twice the number of the training data. Therefore, two input data files, the training data for optimising the parameters and the test data, are read into the testbed.

Each file is organised as records, each of which consists of a vector of attributes x: $x_{1i} = (x_1, x_2, \ldots, x_m)$ followed by the target y: $x_{2i} \in (y_1, y_2, ..., y_c)$ where m is the number of attributes and c is the number of classes. The SVM constructs $c(c - 1)/2$ binary classifiers, and uses the training data to find the optimum separating hyperplane. Finally, the test data is used to compute the classifier accuracy defined by:

$$\text{Acc} = (n / N) * 100,$$

where n is the number of correct classified examples and N is the total number of the test examples.

Table 2 SVM classification accuracy

Data set	*RBF (P, d = 3)*	*POLY (d = 3)*	*RBPF (P, d = 3)*
1	82.6	93.9	93.9
2	85	82.52	90.99
3	94.9	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	90.88	92.98	95.92

Having developed the test bed, the next stage is to compare the classification accuracies of SVMs based on each of the three kernels: POLY, Gauss and the proposed kernel. By determining the classification accuracy for a range of data sets including sets of different number of classes and different numbers of attributes the relative performances of a SVM based on the three kernels can be evaluated. For the range of data sets given in Table 1, the accuracy levels are given in Table 2.

4.3 Result analysis

From Table 2, the Gauss RBF, with p, $d = 3$ gives better classification accuracy with small data sets ($5 \rightarrow 8$) than the POLY function. The POLY kernel, with four dimension data and $d = 3$, gives better results than RBF for large sets ($1 \rightarrow 4$). The proposed kernel, RBPF, gives the best classification accuracy in nearly all the data sets regardless of the size of the data. For data set seven (ABE), although, the size of dataset is small, and one feature is enough to distinguish a pair of ABE classes, the proposed kernel still achieves good results. This is shown by the experimental results given in Table 2. Here $p = 1$, with $d = 3$, and the resulting classification accuracy of the SVM based on the proposed kernel is 99.82%.

When considering the performance of an SVM in terms of classification accuracy, of interest is the relationship between the accuracy and the complexity of the data sets. Ideally, the classification accuracy would be independent on the number of classes and the number of attributes. This relationship is investigated for SVMs based of the three kernels and the results within Tables 1 and 2 are presented in Figures 4–6.

Figure 4 shows the relation between the classification accuracy using the three kernels, RBF, POLY, RBPF with a range of different number of classes. It is clear that the classification accuracy of the SVM based on the proposed kernel (RBPF) is consistently better than the classification accuracy of the SVM based on both RBF and POLY kernels, even in the case of the largest number of classes.

Figure 5 shows the relation between the classification accuracy using the three kernels, RBF, POLY, RBPF with a range of different number of attributes. As with Figure 4, it is clear that the classification accuracy of the SVM based on the proposed kernel (RBPF) is consistently better than the classification accuracy of the SVM based on both RBF and POLY kernels.

Interestingly, Figure 5 also shows that as the number of attributes increases, the improvement in classification accuracy of the SVM based on the proposed kernel compared to the SVMs based on both POLY and RBF kernels also increases. This is clearly seen by comparing the classification accuracies for data set five which has the largest number of attributes. This improved performance is due to the fact that the proposed function is more complex and combines the performance of both its parents, Gauss and POLY functions.

Figure 6 presents the mean classification accuracy obtained for SVMs based on each of the three kernels applied to all eight data sets. The improvement in classification accuracy afford by the SVM based on the proposed kernel is clearly seen as it gives the best mean accuracy compared to either of the SVMs based on the established Gauss or POLY functions.

Figure 4 Classification accuracy of RBF, POLY, RBPF kernels with number of classes (see online version for colours)

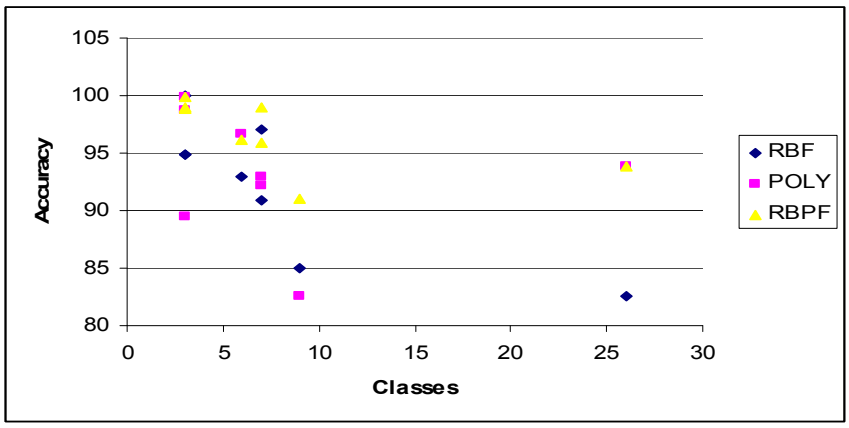


Figure 5 Classification accuracy of RBF, POLY, RBPF kernels with number of attributes (see online version for colours)

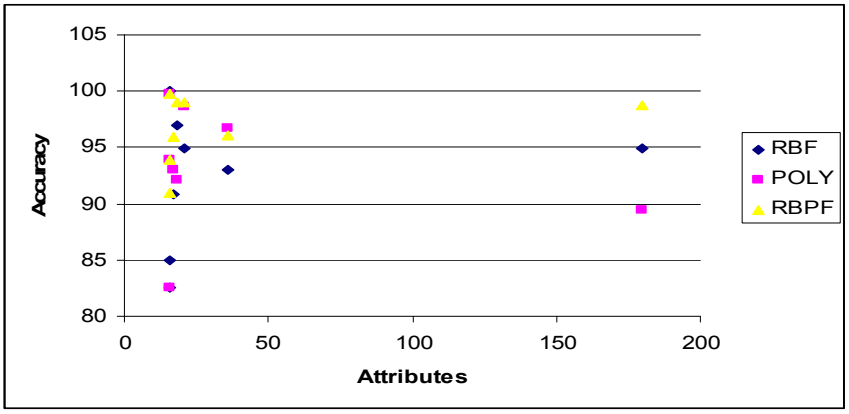
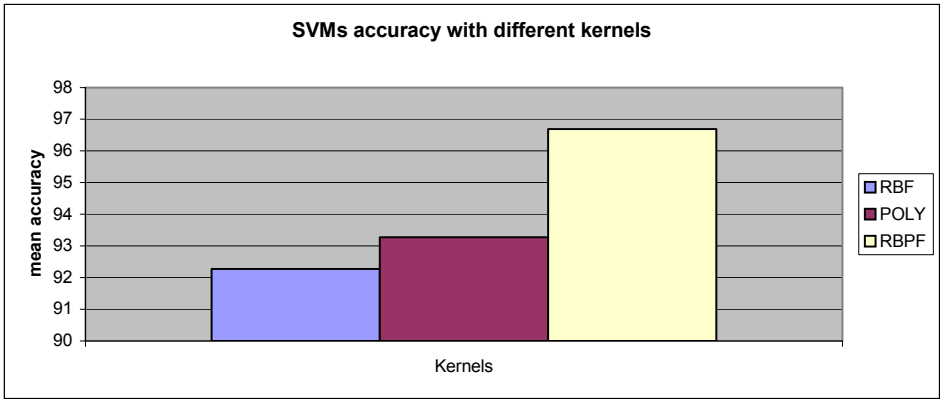


Figure 6 SVM mean classification accuracy (see online version for colours)



5 Conclusions

In this paper, the classification accuracy and numerical efficiency of SVMs have been improved by mapping the training data into a feature space by the aid of new kernel functions. The computational complexity of the classification operation does not depend on the dimensionality of the feature space, which can even be infinite.

Different sizes of data sets have been used with different attributes from two sources (available at http://www.ics.uci.edu/~mlearn/MLRepository.html; http://www.liacc.up.pt/ML/old/statlog/datasets.html). The experimental results show that RBF gives better classification accuracy with small data sets compared to the POLY function. However, the POLY kernel gives better classification results in the large data sets. Whereas the proposed kernel, RBPF, obtains the best accuracy in nearly all the data sets, especially in the largest number of attributes data set, because the proposed function combines the performance of both its parents, Gauss and POLY functions. The results obtained from the data sets are encouraging and suggest that the proposed method is worth further consideration.

References

Boser, B., Guyon, I. and Vapnik, V. (1992) 'A training algorithm for optimal margin classifiers', in Haussler, D. (Ed.): *Annual ACM Workshop on COLT*, Vol. 144, No. 52, ACM Press, Pittsburgh, PA.

Chapelle, O. and Schölkopf, B. (2002) 'Incorporating invariances in non-linear support vector machines', in Dietterich, T.G., Becker, S. and Ghahramani, Z. (Eds.) *Advances in Neural Information Processing Systems*, Vol. 14, pp.594–609, MIT Press, Cambridge, MA.

Cuturi, M. (2007) 'Permanents, transport polytopes and positive definite kernels on histograms', *Proceedings of the 20th International Joint Conference on Artificial Intelligence*.

Haasdonk, B. (2005) 'Feature space interpretation of SVMs with indefinite kernels', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 4.

Hastie, T. and Tibshirani, R. (1998) 'Classification by pairwise coupling', *Annals of Statistics*, Vol. 26, No. 2, pp.451–471.

Hsu, C-W. and Lin, C-J. (2002) 'A comparison of methods for multi-class support vector machines', *IEEE Transactions on Neural Networks*, No. 13, pp.415–425.

Lei, H. and Govindaraju, V. (2005) 'Speeding up multi-class SVM evaluation by PCA and feature selection', *Center for Unified Biometrics and Sensors (CUBS)*.

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

Nashaat, R. (2008) 'Support vector machines for data classification', MSC thesis, submitted to Sohag University.

Ratsch, G. (2005) *A Brief Introduction into Machine Learning*, Friedrich Miescher Laboratory of the Max Planck Society, Germany.

Rifin, R. and Klautau, A. (2004) 'In defense of one vs. all classification', *Journal of Machine Learning Research*, No. 5, pp.101–141.

Sonnenberg, S., Ratsch, G., Schoafer, C. and Scholkopf, B. (2006) 'Large scale multiple kernel learning', *Journal of Machine Learning Research*, Vol. 7, pp.1531–1565.

Tsang, I., Kwok, J. and Cheung, P-M. (2005) 'Core vector machines: fast SVMs training on very large data sets', *Journal of Machine Learning Research*, Vol. 6, pp.271–363.

Williamson, R., Smola, A. and Schölkopf, B. (1999) *Entropy Numbers, Operators and Support Vector Kernels*, pp.44–127, MIT Press, Cambridge, MA.

Zanaty, E.A. and Aljahdali, S. (2008) 'Improving the accuracy of support vector machines', *Proceedings of 23rd International Conference on Computers and their Application*, April, pp.75–83, Cancun, Mexico.