International Journal of Knowledge-based and Intelligent Engineering Systems 11 (2007) 1–18
IOS Press

On optimal degree selection for polynomial kernel with support vector machines: Theoretical and empirical investigations

Shawkat Ali[a,*] and Kate Smith-Miles[b]
[a]*School of Computer Science, Central Queensland University, QLD 4702, Australia VIC 3125, Australia*
[b]*School of Engineering and Information Technology, Deakin University, QLD 4702, Australia VIC 3125, Australia*
E-mail: kate.smith-miles@deakin.edu.au

Abstract.. The key challenge in kernel based learning algorithms is the choice of an appropriate kernel and its optimal parameters. Selecting the optimal degree of a polynomial kernel is critical to ensure good generalisation of the resulting support vector machine model. In this paper we propose Bayesian and Laplace approximation methods to estimate the polynomial degree. A rule based meta-learning approach is then proposed for automatic polynomial kernel and its optimal degree selection. The new approach is constructed and tested on different sizes of 112 datasets with binary class as well as multi class classification problems. An extensive computational evaluation of these methods is conducted, and rules are generated to determine when these approximation methods are appropriate.

Keywords:: Support vector machines, polynomial kernel, rule based method

1. Introduction

Support Vector Machines (SVMs) [7,8,36] are an optimal hyperplane based statistical learning method, which solve classification as well as regression problems. They have been shown to offer improved performance in the areas of bioinformatics [14], text mining [27], fraud detection [15], speaker identification [37] and database marketing [5], among many others. The performance of the SVM method depends however on the suitable selection of a kernel. A kernel is the most important part of the SVM algorithm: it generates the dot products in a higher dimensional feature space. The space could theoretically be of infinite dimension, where linear discrimination is possible. The polynomial and radial basis function (rbf) kernels are the most popular classical SVM kernels. Up to now a good number of kernels have been proposed by researchers, but there is no unique kernel that performs best for all problems. The most common procedure for SVM kernel selection is the trial-and-error approach. Joachims argues that SVMs are universal learners with a simple 'plug-in' of an appropriate kernel function to learn the problems [16]. This is a very lengthy procedure due to a vast range of kernel functions available. Onoda et al. argued that selection of a suitable kernel for SVM is an important research issue for real world applications [25]. A priori kernel selection for SVM is a difficult task for the user though [4,28]. Moreover, we found in the SVM literature [16,24], manually feeding the parametric kernel parameter is a traditional approach for SVM user. Santos and Gomes found polynomial kernel with higher order of the polynomial degree performed better for appearance-based object recognition [30]. But according to Ou et al., polynomial kernel has generalisation difficulties for high ranges of the polynomial degree [26]. Therefore, it is a research issue to automatically select the polynomial kernel function and its optimum degree for SVM.

*Corresponding author. E-mail: s.ali@cqu.edu.au.

ISSN 1327-2314/07/$17.00

Our present research is the first step to provide the solution of these issues of polynomial kernel degree selection for SVM. This work is an extension of our previous research [1] which has focused on automated kernel and kernel parameter selection methods based on data information using a single classical statistical method interquartile range measure. The experiment has done with a small range of classification problems. In present research first, we propose both Bayesian and Laplace methods for estimating the optimal degree of the polynomial kernel. This is a theoretical contribution. We then consider an experimental approach to validate these methods, and to determine when they are most appropriate. We start by classifying 112 problems (see Appendix I) from the UCI Repository [6] and Knowledge Discovery Central [20] database by SVM with polynomial kernel, using a variety of different methods to select the optimal polynomial degree, including our two new methods as well as a trial and error approach. We use 10 fold cross validation for those datasets with fewer than 1000 examples. Otherwise we use the holdout method: 70% for training and the rest for testing. After that we identify the dataset characteristics matrix by using statistical measures following Smith et al. [31,32]. These measures seek to characterise the 112 datasets using a variety of statistical, distance-based and distribution-based measures. All the statistical formulations are available in Matlab statistics toolbox [Statistics toolbox user's guide, 2001]. Finally we use the induction algorithm C5.0 (Windows version See5, http://www.rulequest.com/see5-info.html) to generate the rules to describe which optimal degree estimation method is suitable for which type of problem, given the dataset characteristics and the performance of each method on each dataset. We also examine the rules by 10 Fold Cross Validation (10FCV) performances.

Our paper is organised as follows: In Section 2, we provide some theoretical frameworks regarding SVM and kernel theory, and provide rules for polynomial kernel selection based on our previous work. These rules provide guidelines as to when different kernels, including polynomial, are expected to perform well. Section 3 proposes the two methods for estimating the optimal polynomial degree, Laplace and Bayesian, and measures the performance of these methods on the 112 classification problems with statistical significance test results. All statistical measures used to identify the dataset characteristics matrix are summarized in Section 4. A brief review on rule based learning algorithm C5.0 and the analysis of the experimental results are presented in Section 5, where we also present some rules to describe when the proposed optimal degree estimation methods are appropriate. Finally we conclude our research in Section 6.

2. Support vector machine

Let us consider a dataset D of l independently identically distributed (i.i.d) samples: $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_l, y_l)$. Each sample is a set of feature vectors of length $m, \mathbf{x}_i = \langle x_1, \cdots, x_m \rangle$and the target value$y_i \in \{1, \cdots, k\}$ that represents the multi class membership. Now, the pattern recognition problem or machine learning task is to learn the classes for each pattern by finding a classifier with decision functions $f(\mathbf{x}_i, \alpha_i)$, where $f(\mathbf{x}_i, \alpha_i) = y_i, \alpha_i \in \Lambda, \forall \langle \mathbf{x}_i, y_i \rangle \in D$, and Λ is a set of extract parameters. We consider SVM to learn this problem. SVM learns the problem to estimating the learning parameter by solving the quadratic optimization as follows [39]:

$$\begin{aligned} &\min_{\omega,\xi} \phi(\omega, \xi) = \frac{1}{2} \sum_{m=1}^{k} (\omega_m \cdot \omega_m) + C \sum_{i=1}^{l} \sum_{m \neq y_i}^{\xi_i^m} \\ &\text{subject to: } (\omega_{y_i} \cdot \mathbf{x}_i) + b_{y_i} \geqslant (\omega_m \cdot \mathbf{x}_i) + b_m + 2 - \xi_i^m \\ &\xi_i^m \geqslant 0, \quad i = 1, \cdots, l \quad m \in \{1, \cdots, k\} \setminus y_i \end{aligned} \tag{1}$$

Now we can solve this optimisation problem by finding the saddle point of the Lagrangian:

$$\begin{aligned} L(\omega, b, \xi, \alpha, \beta) = &\frac{1}{2} \sum_{m=1}^{k} (\omega_m \cdot \omega_m) + C \sum_{i=1}^{\ell} \sum_{m=1}^{k} \xi_i^m \\ &- \sum_{i=1}^{\ell} \sum_{m=1}^{k} \alpha_i^m \left[((\omega_{y_i} - \omega_m) \cdot \mathbf{x}_i) + b_{y_i} - b_m - 2 + \xi_i^m \right] - \sum_{i=1}^{\ell} \sum_{m=1}^{k} \beta_i^m \xi_i^m \end{aligned} \tag{2}$$

with the dummy variables: $\alpha_i^{y_i} = 0$, $\beta_i^{y_i} = 0$, $\xi_i^{y_i} = 0$

Table 1
Commonly used SVM kernel functions

Kernel Names	Kernel Functions
linear kernel [Vapnik, 1995]	$K(x_i, x_j) = \left\langle x_i^T x_j \right\rangle$
polynomial kernel [Vapnik, 1995]	$K(x_i, x_j) = \left\langle x_i^T x_j \right\rangle^d$ or $K(x_i, x_j) = \left(\left\langle x_i^T x_j \right\rangle + 1\right)^d$
rbf kernel [Vapnik, 1995]	$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2h^2}\right)$ where $h > 0$
multiquadratic kernel [Evgeniou, et al., 1999]	$K(x_i, x_j) = \left(\|x_i - x_j\|^2 + \tau^2\right)^{\frac{1}{2}}$ where $\tau > 0$
spline kernel [Gunn, 1998]	$K(x_i, x_j) = 1 + \left(x_i^T x_j\right) + \frac{1}{2}\left(x_i^T x_j\right) \min\left(x_i^T x_j\right)^2 - \frac{1}{6}\min\left(x_i^T x_j\right)^3$
sigmoidal kernel [Evgeniou, et al., 1999]	$K(x_i, x_j) = \tanh(\eta\left(x_i^T x_j\right) + \theta)$ where η and θ parameter
Laplace kernel [Ali and Smith, 2004a]	$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|}{h}\right)$ where h is the kernel smoothing parameter.

subject to: $\alpha_i^m \geqslant 0, \beta_i^m \geqslant 0, \xi_i^m \geqslant 0, i = 1, \cdots, \ell$ and $c \in \{1, \cdots, k\} \setminus y_i$

which is maximised with respect to α and β and minimised with respect to ω and ξ by considering the notation:

$$c_i^n = \begin{cases} 1 \; if \; y_i = n \\ 0 \; if \; y_i \neq n \end{cases} \text{ and } A_i = \sum_{m=1}^{k} \alpha_i^m \tag{3}$$

After getting the differentiation, the optimal α is obtained as follows:

$$\alpha_i^o = 2\sum_{i,m} \alpha_i^m + \sum_{i,j,m} \left[-\frac{1}{2} c_j^{y_i} A_i A_j + \alpha_i^m \alpha_j^{y_i} - \frac{1}{2}\alpha_i^m \alpha_j^{y_i}\right] (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{4}$$

Finally the decision function for multiclass SVM is

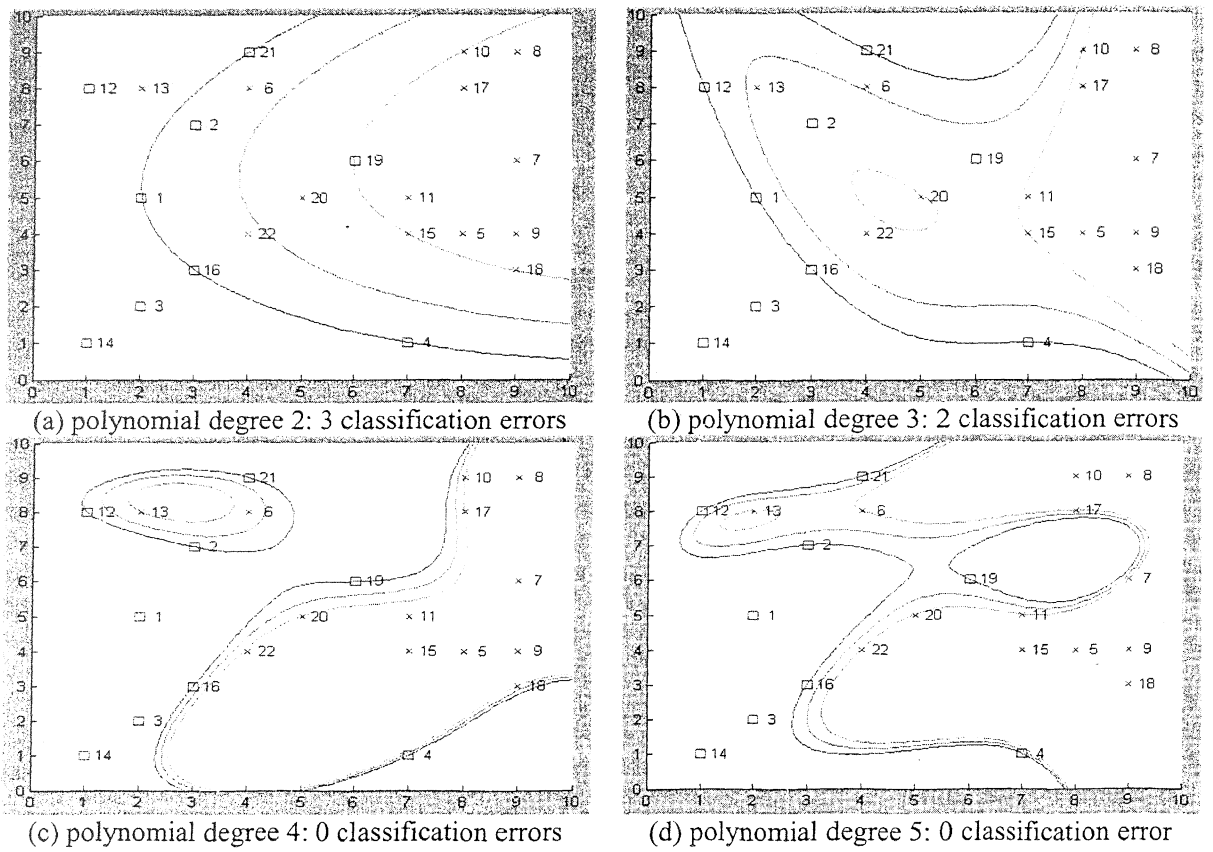
$$\hat{f}(\mathbf{x}) = \arg\max_n \left[\sum_{i:y_i=n} A_i (\mathbf{x}_i \cdot \mathbf{x}) - \sum_{i:y_i \neq n} \alpha_i^n (\mathbf{x}_i \cdot \mathbf{x}) + b_n\right] \tag{5}$$

The inner product $(\mathbf{x}_i \cdot \mathbf{x})$ can be replaced by the convolution inner product $K(\mathbf{x}_i, \mathbf{x}_j)$, also known as the kernel function. Some commonly used SVM kernels with mathematical expressions are listed in Table 1.

Figure 1 shows a pictorial representation of the polynomial kernel with polynomial degree 2–5 for a binary class synthetic problem. The rectangular and the cross sign indicates the two different classes. The middle lines show the Optimal Hyperplane (OH) functions.

We observe in Fig. 1 that polynomial degree 4 and 5 produce the best fit because they classified both classes without error. Figure 1 shows that a higher degree can more easily fit the training data. However, a higher polynomial degree is not always ideal, since overfitting can result in a higher probability of error for predicting classes of future examples. Lawrence and Giles [19] observed on an artificially generated dataset with polynomial degree 2, the approximation is poor. The approximation is reasonably good for degree 10. At the order 20, the approximation function fits the data very well, but the interpolation between training points is very poor due to data overfitting. Data overfitting can also be a very important problem in neural networks, and much work has been devoted to preventing overfitting with techniques, for instance model selection, early stopping, weight decay, and pruning [9, 12,18,19,38]. We examine the overfitting problem for SVM with 'balance-scale' [6] dataset as shown in Fig. 2. We report the performance with 10FCV results. It is clear that, due to overfitting, a higher polynomial degree is not suitable for generalisation. However, the optimal polynomial degree to ensure a good balance between learning and generalisation is dependent on the data to a large extent.

The polynomial kernel function is quite simple compared with some other kernels. The most common practice is to manually evaluate the performance of the kernel with polynomial degree ranging from 2 to 5, and select the best one [16]. We have observed experimentally that the polynomial kernel is the third choice from the listed kernels in Table 1 among a large range of classification problems [2]. But for some specific datasets, polynomial kernel was the best choice for SVM. Now we have two research issues to pursue for polynomial kernel. Firstly, how can we



(a) polynomial degree 2: 3 classification errors

(b) polynomial degree 3: 2 classification errors

(c) polynomial degree 4: 0 classification errors

(d) polynomial degree 5: 0 classification error

Fig. 1. Graphical representation of the polynomial kernel on an artificial dataset with different polynomial degrees. The cross and rectangular sign indicates the two classes of data. The middle lines of the above graphs represent the OH for classification. Those data points located on the margin are called SVs.

know a priori when the polynomial kernel is likely to perform well for a certain dataset, and secondly, how we can find the optimal parameter (degree) for the polynomial kernel.

In our previous study [3], we have found the best rule to describe when the polynomial kernel performs better than the other kernels, based on our empirical evaluation (summarised in Appendix II) as mentioned in Table 2.

```
Rule # 1:  IF (Mahalanobis distance <= 218.276) OR (Z-score <= 0.3913) THEN
we should choose polynomial kernel for SVM classification.
```

For other kernels rules see [3] and the summarised kernels performance have mentioned in Appendix II.

In the following sections we attempt to select the polynomial kernel based on dataset properties. The rule for this kernel is highly acceptable due to higher accuracy rating. We found empirically polynomial kernel showed best classification performance for 34.48% datasets among 112 problems. Now that we have answered the question of when we should select the polynomial kernel, we can turn our attention to the challenge of estimating the optimal degree. In the following section we will examine two different polynomial degree estimation methods, namely Bayesian Information Criterion (BIC) and Laplace approximation based on Principal Component Analysis (PCA). We will present comparative performance results, and then attempt to gain insight into which method should be used for certain datasets.

3. Estimating optimal polynomial degree

We propose modified BIC and Laplace methods to estimate the optimal degree for polynomial kernel. Both methods first estimate the evidence based on dataset information provided by Bayesian probability theory corresponding

Table 2
Confusion matrix based on 10FCV results for the polynomial kernel selection rule on 112 classification problems

Data condition satisfied	Polynomial kernel best	
	Yes(Y)	No(N)
Yes(Y)	2.6	1.4
No(N)	0.8	6.2

Accuracy = 80%.

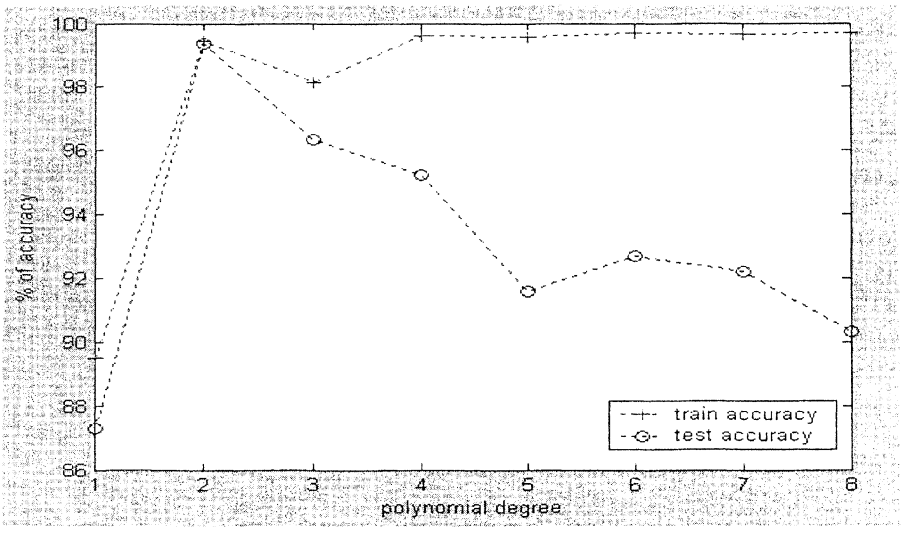


Fig. 2. Data overfitting problem for SVM with polynomial kernel. Individual train and test set performance for 'balance-scale' dataset is shown. Significant overfitting can be shown among degree 2–8 and beyond.

to a range of polynomial degrees. Then the optimal polynomial degree is based on highest evidence that is the best fitness of the polynomial degree. We use this best fit degree as an optimal parameter for SVM polynomial kernel. This section is the summarised form of the formulation for Laplace and BIC methods applied to PCA model based on [17,23,34,35], but modified for our application to polynomial kernel.

Let us consider a d-dimensional vector $\mathbf{x}$ to generate a SVM model from a smaller k-dimensional vector w by a linear transformation with a noise vector e : $\mathbf{x} = Hw + m + e$, where m is the mean of $\mathbf{x}$, while H define its variance. The principal component vector w and the noise variance v are assumed to be spherical Gaussian as follows:

$$p(e) \sim \mathrm{N}(0, v) \quad p(w) \sim \mathrm{N}(0, I_k) \tag{6}$$

Now, following the Gaussian distribution the observation $\mathbf{x}$ becomes:

$$p(\mathbf{x}|H, m, v) \sim \mathrm{N}(m, HH^{\mathrm{T}} + vI) \tag{7}$$

We estimate the basis vectors H and the noise variance v from the dataset$D = \{x_1, \cdots, x_N\}$. The probability distribution of the dataset D is

$$p(D|H, m, v) = (2\pi)^{-Nd/2} \left|HH^T + vI\right|^{-N/2} \exp(-\frac{1}{2} tr((HH^T + vI)^{-1} S)) \tag{8}$$

$$S = \sum_i (\mathbf{x}_i - m)(\mathbf{x}_i - m)^T \tag{9}$$

Now, our aim is to select the subspace dimensionality k for a SVM model. We compute all possible values for this k dimensionality and then finally pick the maximum value of k. First, we define a prior density for all these parameters (m, H, v) by assuming only information contained in dataset D. A non informative prior for m is uniform, and with

such a prior we can integrate out m analytically, leaving

$$p(D|H,v) = N^{-d/2}(2\pi)^{-(N-1)d/2}\left|HH^T + vI\right|^{-(N-1)/2}\exp\left(-\frac{1}{2}tr((HH^T+vI)^{-1}S)\right) \tag{10}$$

where $S = \sum_i (\mathbf{x}_i - \hat{m})(\mathbf{x}_i - \hat{m})^T$, $\hat{m}$ is the maximum likelihood estimated mean.

The basis vector H must have a proper prior since it varies in dimension for different models, unlike m. Let, the decomposed form of H is as follows:

$$H = U(L - vI_k)^{1/2}R \quad U^TU = I_k \quad R^TR = I_k \tag{11}$$

where L is a diagonal matrix with elements l_i. The orthogonal matrix U is the basis, L is the scaling and R is a rotation within the subspace. A conjugate prior for (U, L, R, v), parameterized by α is as follows:

$$p(U,L,H,v) \propto \left|HH^T + vI\right|^{-(\alpha+2)/2}\exp\left(-\frac{\alpha}{2}tr((HH^T+vI)^{-1})\right) \tag{12}$$

The parameter α controls the sharpness of the prior. For a non informative prior, it should be small, making the prior diffuse.

Combining the likelihood with the prior gives

$$p(D|k) = c_k\int_{U,L,v}\left|HH^T + vI\right|^{-n/2}\exp\left(-\frac{1}{2}tr((HH^T+vI)^{-1}(S+\alpha I))\right)dUdLdv \tag{13}$$

where $n = N + 1 + \alpha$

$$c_k = \frac{N^{-d/2}(2\pi)^{-(N-1)d/2}p(U)}{\Gamma((\alpha+2)(d-k))}\left(\frac{\alpha(d-k)}{2}\right)^{(\alpha+2)(d-k)}\frac{1}{\Gamma(\alpha/2)^k}\left(\frac{\alpha}{2}\right)^{\alpha k/2}$$

The likelihood does not consider R, so we just consider a multiplicative factor of $\int_R p(R)dR = 1$.

Laplace's method is a simpler method to integrate Eq. (13), over L and v as follows:

$$\int f(\theta)d\theta \approx f(\hat{\theta})(2\pi)^{rows(A)/2}\left|A\right|^{-1/2} \tag{14}$$

where $\hat{\theta} = \arg\max_\theta f(\theta)$, $A = -\left[\frac{d^2\log f(\theta)}{d\theta_i d\theta j}\right]_{\theta=\hat{\theta}}$.

Now, the key objective is a good approximation for the parameter $\theta = (U, L, v)$. Due to the positive scale parameters l_i and v, we can consider $l_i' = \log(l_i)$ and $v' = \log(v)$. Therefore,

$$\hat{l}_i = \frac{N\lambda_i + \alpha}{N-1+\alpha} \quad \hat{v} = \frac{N\sum_{j=k+1}^{d}\lambda_j}{n(d-k)-2} \tag{15}$$

$$\left.\frac{d^2\log f(\theta)}{(dl_i')^2}\right|_{\theta=\hat{\theta}} = -\frac{N-1+\alpha}{2} \quad \left.\frac{d^2\log f(\theta)}{(dv')^2}\right|_{\theta=\hat{\theta}} = -\frac{n(d-k)-2}{2} \tag{16}$$

where λ is the eigenvalue. The dimension of the orthogonal matrix U is $m = dk - k(k+1)/2$, since we are imposing $k(k+1)/2$ constraints on a $d \times k$ matrix. The prior density of U is

$$p(U) = 2^{-k}\prod_{i=1}^{k}\Gamma((d-i+1)/2)\pi^{-(d-i+1)/2} \tag{17}$$

The matrix U could be parameterized by using Euler vector representation as follows:

$$U = U_d\exp(Z)\begin{bmatrix} I_k \\ 0 \end{bmatrix} \tag{18}$$

where U_d is a fixed orthogonal matrix and Z is a skew-symmetric matrix of parameters.

Now, the integrated function for U is

$$p(U|D,L,v)\infty \exp\left(-\frac{1}{2}tr((L^{-1}-v^{-1}I)U^{T}SU)\right) \tag{19}$$

The density is maximized when U contains the top k eigenvectors of S. If we consider to be the top d eigenvectors of S:

$$U_d^T S U_d = N\Lambda \tag{20}$$

where $\hat{\Lambda} = \begin{bmatrix} L & 0 \\ 0 & \hat{\nu} I_{d-k} \end{bmatrix}$.

Now by considering $dz_{ji} = -dz_{ij}$, we get

$$d^2 \log f(\theta)\big|_{z=o} = -\sum_{i=1}^{k}\sum_{j=i+1}^{d}\left(\hat{\lambda}_j^{-1}-\hat{\lambda}_i^{-1}\right)(\lambda_i-\lambda_j)\,N dz_{ij}^2 \tag{21}$$

Since the Hessian matrix is diagonal, the second derivative is as follows:

$$|A_Z| = \prod_{i=1}^{k}\prod_{j=i+1}^{d}\left(\hat{\lambda}_j^{-1}-\hat{\lambda}_i^{-1}\right)(\lambda_i-\lambda_j)\,N \tag{22}$$

In Eq. (22) A is a block diagonal and $|A| = |A_Z|\,|A_L|\,|A_v|$. We know A_z from Eq. (22), A_L and A_v from Eq. (16). Finally replacing these values in Eq. (14) we get the evidence:

$$p(D|k) \approx 2^k c_k \left|\hat{L}\right|^{-n/2} \hat{v}^{-n(d-k)/2} e^{-nd/2} (2\pi)^{(m+k+1)/2} |A_Z|^{-1/2} |A_L|^{-1/2} |A_v|^{-1/2} \tag{23}$$

For a polynomial model, the optimal degree is offer by k in the Laplacian approximation with small value of α and reasonably large N:

$$p(D|k) \approx p(U)\left(\prod_{j=1}^{k}\lambda_j\right)^{-N/2} \hat{v}^{-N(d-k)/2}(2\pi)^{(m+k)/2}|A_Z|^{-1/2}N^{-k/2} \tag{24}$$

where, $\hat{l}_i = \lambda_i$, $\hat{v} = \frac{\sum_{j=k+1}^{d}\lambda_j}{d-k}$.

Equation (24) offers the evidence for the best suitable polynomial degree, which we may call degree of best polynomial fitness.

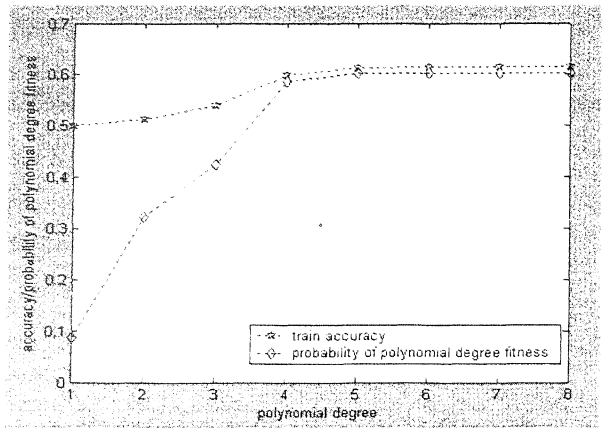
A simplified implementation of the Laplacian optimal polynomial degree approximation is BIC as follows:

$$p(D|k) \approx \left(\prod_{j=1}^{k}\lambda_j\right)^{-N/2} \hat{v}^{-N(d-k)/2} N^{-(m+k)/2} \tag{25}$$

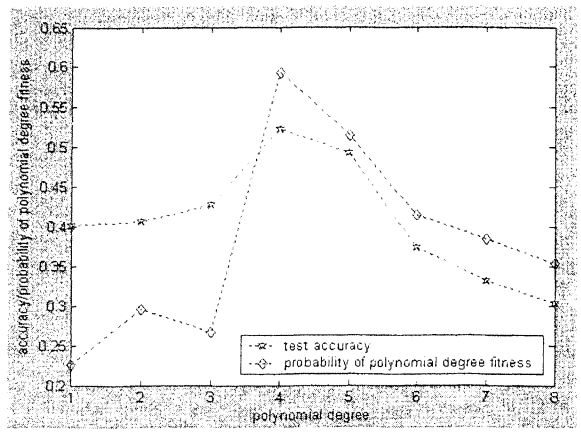
Equations (24) and (25) offer different methods for finding the optimal degree k (or d as shown in Table 1). These two probability functions can be readily evaluated for different values of k in an iterated procedure which is not very computationally expensive. The resulting probabilities provide an estimate of the likely fitness of different polynomial kernel degrees and the maximum fitness can be selected as an estimate of the optimal degree.

A pictorial view of this idea for dataset ann1 with Laplace method for optimal degree estimation and its corresponding classification performance is shown separately for train and test dataset as in Fig. 3.

In Fig. 3, the Laplace method predicts the optimal degree is 4 based on higher evidence (highest probability of polynomial degree fitness) for ann1 dataset. We observe the performance of the training set increases with higher order of polynomial degree. On the other hand, the performance of the test set decreases with the higher order of polynomial degree. This is clear evidence of data overfitting and reinforces the need for methods to select the optimal degree.



(a) Comparison of estimated (Laplace) accuracy and actual accuracy on ann1 training dataset.



(b) Comparison of estimated (Laplace) accuracy and actual accuracy on ann1 test dataset.

Fig. 3. Optimum polynomial degree selection by Laplace method and corresponding classification performance is shown separately for train and test set with ann1 dataset.

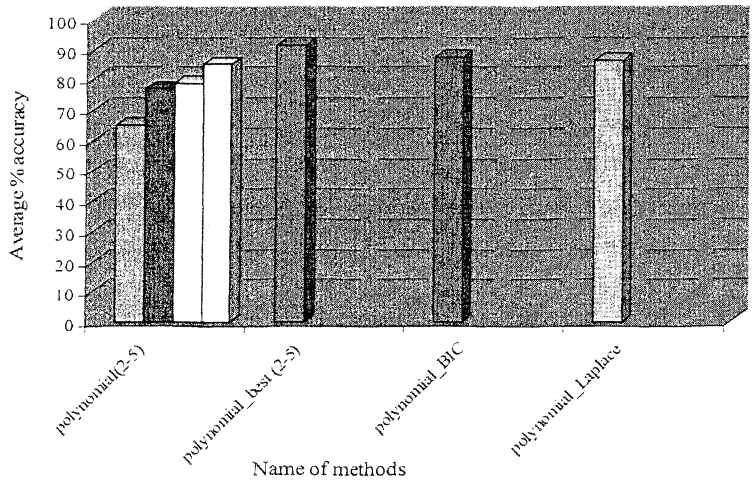


Fig. 4. Average test set accuracy for different polynomial kernel parameter fitting methods for problems satisfying rule # 1 (39 datasets).

3.1. Optimal degree estimation performance: Accuracy

The average test set classification performance of polynomial kernel with parameter 2–5, polynomial best (best performance manually selected from degree 2–5), optimum polynomial degree approximation by BIC and Laplace methods is shown in Fig. 4. The results are presented only for those 39 of the 112 original datasets that are suited to the polynomial kernel (satisfy rule #1).

The BIC and Laplace methods showed close performance with the optimal polynomial accuracy found through exhaustive search of degree range 2 to 5. Both methods showed average higher accuracy than individual polynomial degree 2–5 performance and performance for large dataset (more than 1000 samples) was better than polynomial best performance. For large datasets polynomial best showed average 72.39%, BIC and Laplace methods showed 73.21%. BIC method predicted the optimal degree for polynomial kernel for 53.85% of the datasets where polynomial kernel is expected to be best. On the other hand Laplace method predicted the optimal degree for 51.28% of the datasets. We observed that 39.28% of the datasets have optimal degrees outside the range of 2 to 5. For many of the datasets BIC and Laplace methods predicted the same polynomial degree among the 112 problems. The polynomial kernel performance with datasets better suited to others (non-polynomial) kernel is showed in Fig. 5.

Table 3
Average computational performance for different polynomial kernel parameter estimation methods

Average computational	Polynomial_Best	Polynomial_BIC	Polynomial_Laplace
time in Sec.	819.67	0.0185	0.0488

Table 4
Results of the t-test for all methods of polynomial degree selection

Algorithms	Hypothesis H	Significance P	Confidence Interval CI	
polynomial_best vs polynomial_2	1	1.2321e-005	0.1578	0.3707
polynomial_ best vs polynomial_3	1	6.2817e-004	0.0665	0.2247
polynomial_ best vs polynomial_4	1	8.5060e-004	0.0559	0.1975
polynomial_ best vs polynomial_5	1	0.0019	0.0256	0.1046
polynomial_ best vs polynomial_BIC	0	0.4378	−0.0644	0.1458
polynomial_ best vs polynomial_Laplace	0	0.3616	−0.0602	0.1612

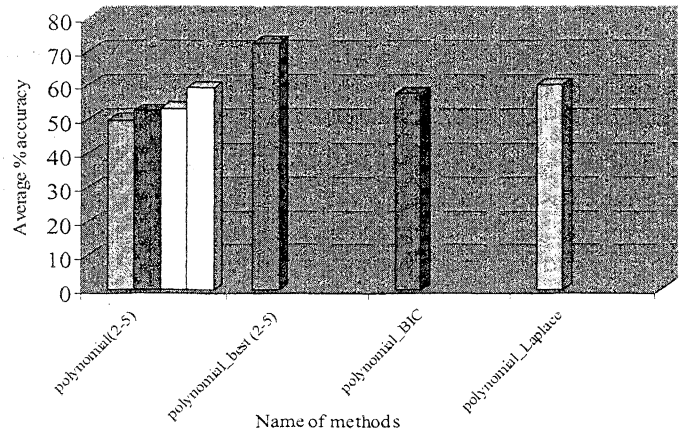


Fig. 5. Average test set accuracy for different polynomial kernel parameter fitting methods for problems not satisfying rule # 1 (73 datasets).

3.2. Optimal degree estimation performance: Computational time

The computational performance to determine the best polynomial degree using the three methods: polynomial best (exhaustive search of degree 2–5) and estimation by BIC and Laplace methods, is shown in Table 3.

The exhaustive optimal degree search method needed extremely higher computational time than BIC and Laplace methods. It selects the polynomial degree one by one from a range 2–5 to train the SVM polynomial model. But both BIC and Laplace methods estimate the optimal polynomial degree for SVM by a simple iteration of Eqs (25) and (24) respectively that estimates the likely performance of the SVM model without the need to build such models. Therefore, BIC and Laplace methods show the superior computational performance compared to exhaustive search method.

3.3. Significance test

The t-test results are summarised in Table 4. We considered the base kernel as polynomial best. The test input was the percentage of correct classification of test set of all the methods for manual and optimal polynomial degree selection.

The outputs of $H = 0$ in the above table indicates we may not reject the null hypothesis that there is no significant difference in results. Alternatively, $H = 1$ means we may reject the null hypothesis. The polynomial best (exhaustive search of polynomial degree 2–5) showed significant performance difference with the individual models. In other words, a much better final result is obtained by the trial-and-error approach compared to selecting just

one model, as one would expect. But, the polynomial best compared to BIC and Laplace methods showed no significant performance difference in accuracy. The higher values of the significance level suggested accepting the null hypothesis. The BIC and Laplace methods give results comparable to exhaustive search, but are much faster to implement.

The average percentage of classification performance and significance testing has shown that classification accuracy depends on particular polynomial kernel degree selection. A detailed polynomial optimal degree estimation performance by BIC and Laplace methods is represented in Appendix III. We observe from both of these optimal polynomial degree estimation methods, the optimal degree is frequently more than 5. Any single method is not always best to estimate the optimal polynomial degree. So, we need a method to provide a priori information about which optimal degree estimation method is suitable for which classification problem with SVM.

In the following section we describe the methodology we use to assist in the appropriate selection of an optimal degree estimation method for a given dataset. First each dataset is described by a set of measurable meta characteristics; we then combine this information with the performance results; and finally use a rule-based induction method to provide rules describing when each optimal parameter estimation method for polynomial kernel is likely to perform well.

4. Datasets characteristics measurement

Each dataset can be described by simple, distance and distribution based statistical measures [31,32]. These three sets of measures characterise the datasets in different ways. First, the simple classical statistical measures identify the data characteristics based on variable to variable comparisons. Then, the distance based measures identify the data characteristics based on sample to sample comparisons. Finally, the density based measures consider the single data point from a matrix to identify the datasets characteristics. We average most of statistical measures over all the variables and take these as global measures of the dataset characteristics. In statistics, basically central tendency measures are use for the location of the middle or the center of a data distribution. The term center is purposely left somewhat vague so that the term central tendency can refer to a wide variety of measures. However, the mean is the most commonly used measure of central tendency. Therefore we use mean for data characteristics measurements. The simple statistical measures are calculated within each column, and then averaged over all columns to obtain global measures of the dataset. Likewise, the distance measures are averaged over all pairwise comparisons, and the density based measures are averaged across the entire matrix.

4.1. Simple statistical measures

Descriptive statistics can be used to summarise any large dataset into a few numbers that contain most of the relevant characteristics of that dataset. The following table lists the statistical measures used in this work as provided by the Matlab Statistics Toolbox and some other different sources [22] as follows:

Meta Attribute Names	Meta Attribute Names
Geometric mean	Max. and Min. eigenvalue
Harmonic mean	Skewness
Trim mean	Kurtosis
Standard deviation	Correlation Coefficient
Interquartile Range	Prctile

4.2. Distance based measures

Distance based measures calculate the dissimilarity between samples. We measure the euclidean, city block and mahalanobis distance between each pair of observations for each dataset as follows:

Meta Attribute Names	Meta Attribute Names
Euclidean distance	Mahalanobis distance
City Block distance	

4.3. Distribution based measures

The probability distribution of a random variable describes how the probabilities are distributed over the various values that the random variable can take on. We measure the probability density function (pdf) and cumulative distribution function (cdf) for all datasets by considering different types of distributions as follows:

Meta Attribute Names	Meta Attribute Names
Chi-square pdf	Chi-square cdf
Normal pdf	Normal cdf
Binomial pdf	Discrete uniform cdf
Exponential pdf	F pdf
Gamma pdf	Hypergeometric cdf
Lognormal pdf	Poisson pdf
Rayleigh pdf	Student's t pdf

These measures are all calculated for each of the datasets to produce a dataset characteristics matrix. Finally by combining this matrix with the performance results in Appendix III, we can derive rules to suggest when certain optimal degree estimation methods are appropriate.

5. Rule generation

The trial-and-error approach is a very common procedure to select the optimal degree for polynomial kernel. It is a computationally complex task to find the best degree by following this procedure. If we are interested in applying a specific method to a particular problem we have to consider which method is more suitable for which problem. The suitability test can be done from rules developed with the help of the data characteristics properties.

Rule based learning algorithms, especially decision trees (also called classification trees or hierarchical classifiers), are a divide-and-conquer approach or a top-down induction method, that have been studied with interest in the machine learning community. Quinlan [29] introduced the C4.5 and C5.0 algorithms to solve classification problems. C5.0 works in three main steps. First, the root node at the top node of the tree considers all samples and passes them through to the second node called 'branch node'. The branch node generates rules for a group of samples based on an entropy measure. In this stage C5.0 constructs a very big tree by considering all attribute values and finalises the decision rule by pruning. It uses a heuristic approach for pruning based on statistical significance of splits. After fixing the best rule, the branch nodes send the final class value in the last node called the 'leaf node' [10, 29]. C5.0 has two parameters: the first one is called the pruning confidence factor (c) and the second one represents the minimum number of branches at each split (m). The pruning factor has an effect on error estimation and hence the severity of pruning the decision tree. The smaller value of c produces more pruning of the generated tree and a higher value results in less pruning. The minimum branches m indicates the degree to which the initial tree can fit the data. Every branch point in the tree should contain at least two branches (so a minimum number of $m = 2$. For detail formulations see [29].

Now that the characteristics of each dataset can be quantitatively measured, we can combine this information with the empirical evaluation of kernel parameter estimation method performance and construct the dataset characteristics matrix. Thus, the result of the jth degree selection method on the ith dataset is calculated as:

$$R_{ij} = 1 - \frac{e_{ij} - \max(e_i)}{\min(e_i) - \max(e_i)} \tag{26}$$

where e_{ij} is the percentage of correct classification for the jth method on dataset i, and e_i is a vector of accuracy for dataset i. The class values in the matrix are assigned based on the performance best rank. The best rank is defined as 1 and the worst is 0. For example, if BIC method shows the ranking performance 1 for the dataset A, then the class

Table 5
Confusion matrix based on 10FCV results for the BIC method selection rule

Data condition satisfied	BIC method best	
	Yes(Y)	No(N)
Yes(Y)	2.0	0.1
No(N)	0.2	1.6

Accuracy = 92.31%.

Table 6
Confusion matrix based on 10FCV results for the Laplace method selection rule

Data condition satisfied	Laplace method best	
	Yes(Y)	No(N)
Yes(Y)	2.0	0
No(N)	0.4	1.5

Accuracy = 89.74%.

in the matrix for problem A is BIC. Based on the 112 classification problems we can then train a rule-based classifier (C5.0) to learn the relationship between dataset characteristics and degree selection method performance. We split the matrix 90% to construct the model tree. The process is then repeated using a 10 fold cross validation approach so that 10 trees are constructed. From these 10 trees, the best rules are found for each optimal degree selection method based on the best test set results. The generalisation of these rules is then tested by applying each of the randomly extracted test sets and calculating the average accuracy of the rules as discussed below in Tables 5 and 6. We found the suitable parameter value for global pruning factor; c is 70–90% and the number of minimum branches m is 2.

We have demonstrated the rules for polynomial kernel in Section 2. Now if any dataset satisfies the polynomial kernel rule then we need to find optimal polynomial degree estimation method. So, in the following section we will generate the rules describing when to choose the BIC and Laplace methods for optimal polynomial degree estimation.

5.1. Rules for BIC method

The best rules for BIC method is generated with $c = 90\%$ and $m = 2$ as follows:

```
Rule # 2:  IF (skewness > 1.2843) OR (exponential pdf <= 19.2502), THEN we
should choose BIC method for polynomial kernel degree estimation.
```

5.2. Rules for Laplace method

The best rules for Laplace method is generated with c = 70% and m = 2 as follows:

```
Rule # 3:  IF (median > 2.5 AND normal pdf <= 0.15851) OR (median > 2.5 AND
gamma pdf <= 8.9639e-006 AND student's t pdf <= 18.0492) THEN we should
choose Laplace method for polynomial kernel degree estimation.
```

The generated rules show around 90% accuracy. Individually we observed BIC approximation method showed slightly better performance than Laplace approximation, although Laplace was superior for problems not best suited to polynomial kernel as shown in Fig. 5. These rules might be useful to determine which polynomial degree approximation method is most appropriate for which problem.

6. Conclusions

In this research we have widely investigated both theoretically and empirically how to select polynomial kernel and its optimal degree for SVM. We proposed a simple rule for polynomial kernel and optimal degree selection based on dataset information. This method is much faster than trial-and-error based selection of the polynomial degree. Since both proposed polynomial degree estimation methods are based on Bayesian information, this approximation works especially well for large datasets with more than 1000 samples. We have observed that the best polynomial degree is commonly out of the range 2 to 5 (the common range tested in the literature). The estimated higher degree increased the kernel performance accuracy for some specific cases. The BIC and Laplace methods are very fast to estimate the optimal polynomial degree. Datasets when BIC and Laplace methods perform poorly, are the same as those for which polynomial kernel is not recommended by rule # 1. We examined the generated rules by 10FCV. All generated rules shown high accuracy ratings. We suggest the default polynomial parameter setting is the traditional approach, meaning if any dataset satisfies the polynomial rule but does not satisfy the BIC or Laplace rules then we should manually try polynomial degree from 2–5. The main benefit of our methodology is that we can achieve higher accuracy for some classification problems and significant savings in time by understanding the characteristics of the dataset. We have planned to investigate on-line parameter setting for SVM as follows [21] with polynomial kernel.

Acknowledgements

The authors are grateful to the suggestions of the two anonymous reviewers and associate editor which greatly improved the paper.

References

[1] S. Ali and K.A. Smith, *Automatic parameter selection for polynomial kernel,* Proceedings of the IEEE International Conference on Information Reuse and Integration, 2003, 243–249.
[2] S. Ali and K.A. Smith, Laplace kernel with automatic smoothing parameter estimation for support vector machine, Submitted in *Journal of Computational Management Science,* Springer-Verlag Berlin, Heidelberg, 2004.
[3] S. Ali and K.A. Smith, Automatic kernel selection for support vector machines, Accepted for *International Journal of Neurocomputing*, Elsevier Scienc., 2004.
[4] S.-I. Amari and S. Wu, Improving Support Vector Machine Classifiers by Modifying Kernel Functions, *Neural Networks* **12** (1999), 783–789.
[5] K.P. Bennett, S. Wu and L. Auslender, *On support vector decision trees for database marketing,* IEEE International Joint Conference on Neural Networks (IJCNN '99), 1999, 2, 904–909.
[6] C. Blake and C.J. Merz, UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html, Irvine, CA: University of California, 2002.
[7] B.E. Boser, I. Guyon and V.N. Vapnik, *A Training algorithm for optimal margin classifiers,* in proceedings of the Fifth Annual Workshop of Computational Learning Theory, Pittsburg, ACM 5, 1992, 144–152.
[8] C. Cortes and V. Vapnik, Support vector networks, *Machine Learning* **20** (1995), 273–297.
[9] Y.L. Cun, J.S. Denker and S.A. Solla, Optimal brain damage, in: *Advances in Neural Information Processing Systems,* D.S. Touretzky, eds, San Mateo, Morgan Kaufmann, San Mateo, CA, 1990, 2, 598–605.
[10] R.P.W. Duin, A note on comparing classifier, *Pattern Recognition Letters* **1** (1996), 529–536.
[11] T. Evgeniou, M. Pontil and T. Poggio, Regularization Networks and Support Vector Machines, *Advances in Computational Mathematics* **13**(1) (2000), 1–50.
[12] S. Geman, E. Bienenstock and R. Doursat, Neural networks and the bias/variance dilemma, *Neural Computation* **4**(1) (1992), 1–58.
[13] S.R. Gunn, Support vector machines for classification and regression, ISIS Technical Report, Image Speech and Intelligent Systems Group, University of Southampton, UK, 1998.
[14] I. Guyon, J. Weston, S. Barnhill and V. Vapnik, Gene selection for cancer classification using support vector machines, *Machine Learning* **46**(1/3) (2002), 389–422.
[15] K. Hyun-Chul, P. Shaoning, J. Hong-Mo, K. Daijin and B. Sung-Yang, *Pattern classification using support vector machine ensemble,* Proceedings of IEEE 16th International Conference on Pattern Recognition, 2002, 2, 160–163.
[16] T. Joachims, Text categorization with support vector machines: learning with many relevant features, in: *Proceedings of ECML-98, 10th European Conference on Machine Learning,* C. N'edellec and C. Rouveirol, eds, 1998, pp. 137–142.
[17] R.E. Kass and A.E. Raftery, Bayes factors and model uncertainty. Technical Report 254, University of Washington, 1993.

[18] A. Krogh and J.A. Hertz, A simple weight decay can improve generalization, in: *Advances in Neural Information Processing Systems,* J.E. Moody et al., eds, Morgan Kaufmann, San Mateo, CA, 1992, 4, 950–957.
[19] S. Lawrence and C.L. Giles, *Overfitting and neural networks: conjugate gradient and backpropagation,* in proceedings of the IEEE international joint conference on Neural Networks, 2000, 114–119.
[20] T.-S. Lim, Knowledge discovery central, Datasets, 2002, http://www.KDCentral.com/.
[21] J. Ma, J. Theiler and S. Perkins, Accurate online support vector regression, *Neural Computation* **15** (2003), 2683–2703.
[22] W. Mandenhall and T. Sincich, *Statistics for Engineering and the Sciences,* 4th ed., Prentice Hall, 1995.
[23] T. Minka, Automatic choice of dimensionality for PCA. Technical Report 514, MIT Media laboratory perceptual computing section, 1999.
[24] K. Morik, P. Brockhausen and T. Joachims, *Combining statistical learning with a knowledge-based approach – a case study in intensive care monitoring,* in Proc. 16th International conf. on Machine Learning, 1999, 268–277.
[25] T. Onoda, H. Murata, G. Ratsch and K.-R. Muller, *Experimental Analysis of Support Vector Machines with Different Kernels Based on Non-Intrusive Monitoring Data,* IEEE Proceedings of the 2002 International Joint Conference on Neural Networks, 2002, 3, 2186–2191.
[26] Y.-Y. Ou, C.-Y. Chen, S.-C. Hwang and Y.-J. Oyang, *Expediting model selection for support vector machines based on data reduction,* IEEE International Conference on Systems, Man and Cybernetics, 2003, 1, 786–791.
[27] G. Paab, E. Leopold, M. Larson, J. Kindermann and S. Eickeler, *SVM classification using sequences of phonemes and syllables,* Proceedings of European Conference on Machine Learning (ECML), Helsinki, 2002.
[28] E. Parrado-Hernandez, I. Mora-Jimenez, J. Arenas-Garca, A.R. Figueiras-Vidal and A. Navia-Vazquez, Growing support vector classifiers with controlled complexity, *Pattern Recognition* **36** (2003), 1479–1488.
[29] R. Quinlan, *C4.5: Programs for Machine Learning,* Morgan Kaufman Publishers, San Mateo, CA, 1993.
[30] E.M.D. Santos and H.M. Gomes, *A Comparative study of polynomial kernel SVM applied to appearance-based object recognition,* Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines, Lecture Notes in Computer Science, Springer-Verlag, London, 2002, 408–418.
[31] K.A. Smith, F. Woo, V. Ciesielski and R. Ibrahim, Modelling the relationship between problem characteristics and data mining algorithm performance using neural networks, in: *Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Complex Systems,* C. Dagli et al., eds, ASME Press, 2001, 11, pp. 357–362.
[32] K.A. Smith, F. Woo, V. Ciesielski and R. Ibrahim, Matching data mining algorithm suitability to data characteristics using a self-organising map, in: *Hybrid Information Systems,* A. Abraham and M. Koppen, eds, Physica-Verlag, Heidelberg, 2002, pp. 169–180.
[33] Statistics toolbox user's guide, Version 3, The MathWorks, Inc. USA, 2001.
[34] M.E. Tipping and C.M. Bishop, Mixtures of probabilistic principal component analysers, *Neural Computation* **11**(2) (1999), 443–482.
[35] M.E. Tipping and C.M. Bishop, Probabilistic principal component analysis, *Journal of Royal Statistical Society* **61**(3) (1999), 611–622.
[36] V. Vapnik, *The Nature of Statistical Learning Theory,* Springer-Verlag, New York, 1995.
[37] V. Wan and S. Renals, *Evaluation of kernel methods for speaker verification and identification,* Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '02), 2002, 1, 669–672.
[38] A.S. Weigend, D.E. Rumelhart and B.A. Huberman, Generalization by weight-elimination with application to forecasting, in: *Advances in Neural Information Processing Systems,* R.P. Lippmann et al., eds, Morgan Kaufmann, San Mateo, CA, 1991, 3, 875–882.
[39] J. Weston and C. Watkins, Multi-class support vector machines, in: *Proceedings of 7th European Symposium on Artificial Neural Networks (ESANN99),* M. Verleysen, eds, Bruges, Belgium, 1999.

Appendix I: Datasets description

# Datasets	Dataset names	# samples	# attributes	# classes	# Datasets	Dataset names	# samples	# attributes	# classes
1	abalone	1253	8	3	57	mushroom	1137	11	2
2	adp	1351	11	3	58	musk1	476	166	2
3	adult+stretch	20	4	2	59	musk2	1154	15	2
4	adult-stretch	20	4	2	60	nettalk_stress	1141	7	5
5	allbp	840	6	3	61	new-thyroid	215	5	3
6	ann1	1131	6	3	62	page-blocks	1149	10	5
7	ann2	1028	6	3	63	pendigits-8	1399	16	2
8	aph	909	18	2	64	pha	1070	9	5
9	art	1051	12	2	65	phm	1351	11	3
10	australian	690	14	2	66	phn	1500	9	2
11	balance-scale	625	4	3	67	pid	532	7	2
12	bcw	699	9	2	68	pid_noise	532	15	2
13	bcw_noise	683	18	2	69	pima	768	8	2
14	bld	345	6	2	70	poh	527	11	2
15	bld_noise	345	15	2	71	post-operative	90	8	3
16	bos	910	13	3	72	primary-tumor	339	17	2
17	bos_noise	910	25	3	73	pro	1257	12	2
18	breast-cancer	286	6	2	74	promoter	106	57	2
19	breast-cancer-wisconsin	699	9	2	75	pvro	590	18	2
20	bupa	345	6	2	76	rph	1093	8	2
21	c	1500	15	2	77	shuttle-landing-control	15	6	2
22	cleveland-heart	303	13	5	78	sick-euthyroid	1582	15	2
23	cmc	1473	9	3	79	sma	409	7	4
24	cmc_noise	1473	15	3	80	smo	1429	8	3
25	crx	490	15	2	81	smo_noise	1299	15	3
26	dar	1378	9	5	82	sonar	208	60	2
27	dhp	1500	7	2	83	splice	1589	60	3
28	dna	2000	60	3	84	switzerland-heart	123	8	5
29	dna_noise	2000	80	3	85	t_series	62	2	2
30	DNA-n	1275	60	3	86	tae	151	5	3
31	dph	590	10	2	87	tae_noise	151	10	3
32	echocardiogram	131	7	2	88	thy_noise	3772	35	3
33	flare	1389	10	2	89	tic-tac-toe	958	9	2
34	german	1000	24	2	90	titanic	2201	3	2
35	glass	214	10	6	91	tmris	100	3	2
36	hayes-roth	160	5	3	92	tqr	1107	11	2
37	h-d	303	13	2	93	trains-transformed	10	16	2
38	hea	270	13	2	94	ttt	958	9	2
39	hea_noise	270	20	2	95	va-heart	200	8	4
40	heart	270	13	2	96	veh	846	18	4
41	hepatitis	155	19	2	97	veh_noise	761	30	4
42	horse-23	368	22	2	98	vot_noise	391	30	2
43	horse-colic	368	27	2	99	wdbc	569	30	2
44	house-votes-84	435	16	2	100	wine	178	13	3
45	ionosphere	351	33	2	101	wpbc	199	33	2
46	iris	150	4	3	102	xaa	94	18	4
47	khan	1063	5	2	103	xab	94	18	4
48	labor-neg	40	16	2	104	xac	94	18	4
49	lenses	24	5	3	105	xad	94	18	4
50	letter-a	1334	16	2	106	xae	94	18	4
51	lung-cancer	32	56	2	107	xaf	94	18	4
52	lymphography	148	18	8	108	xag	94	18	4
53	mha	1269	8	4	109	xah	94	18	4
54	monk1	556	6	2	110	xai	94	18	4
55	monk2	601	6	2	111	yha	1601	9	2
56	monk3	554	6	2	112	zoo	101	16	7

Appendix II: Kernels Performance for The Test Data Sets (% Accuracy)

kernel	polynomial				rbf						Laplace
	2	3	4	5	0.2	0.4	0.6	0.8	1.0	1.20	EM
Mean	55.89	58.46	59.58	63.49	70.28	70.17	68.71	67.05	66.01	65.15	70.54
Standard Deviation	23.50	24.34	22.70	22.52	21.06	21.63	22.61	23.31	23.83	23.99	21.04

kernel	spline	multiquadratic					sigmoidal				
		0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1
Mean	54.77	49.62	49.89	50.30	50.90	51.68	61.85	62.85	63.37	63.37	63.49
Standard Deviation	22.86	24.96	24.79	24.64	24.41	24.07	17.34	17.57	17.79	17.90	18.10

Appendix III: Optimal Degree Performance for Different Methods Based on Test Dataset

Datasets name	Manual search method (2–5)		BIC method		Laplace method	
	test set R_{ij}	best degree	test set R_{ij}	estimated optimal degree	test set R_{ij}	estimated optimal degree
abalone	0.99	2	0.60	7	0.60	7
adp	0.86	2	0.87	10	0.87	10
adult+stretch	1.00	2	1.00	3	1.00	3
adult-stretch	1.00	2	1.00	3	1.00	3
allbp	0.96	5	0.96	5	0.96	5
ann1	0.52	4	0.49	5	0.49	5
ann2	0.03	5	0.03	5	0.03	5
aph	1.00	3	0.83	17	0.83	17
art	1.00	4	0.91	11	0.91	11
australian	0.71	5	1.00	13	1.00	13
balance-scale	1.00	2	0.97	3	0.97	3
bcw	0.97	3	0.96	8	0.96	8
bcw_noise	0.95	4	0.83	17	0.83	17
bld	0.84	2	0.49	5	0.49	5
bld_noise	0.69	2	0.34	14	0.46	6
bos	0.82	5	0.61	12	0.61	12
bos_noise	0.93	5	0.89	24	0.89	24
breast-cancer	0.57	2	0.08	5	0.18	4
breast-cancer-wisconsin	0.97	4	0.96	8	0.96	8
bupa	0.74	2	0.48	5	0.48	5
c	1.00	2	1.00	2	1.00	2
cleveland-heart	0.43	5	1.00	12	0.94	10
cmc	0.93	2	0.52	8	0.52	8
cmc_noise	0.71	5	1.00	14	1.00	14
crx	0.65	5	1.00	14	1.00	13
dar	1.00	5	0.88	8	0.88	8
dhp	0.91	5	0.88	6	0.88	6
dna	1.00	5	0.93	8	0.93	8
dna_noise	1.00	5	1.00	5	1.00	5
DNA-n	1.00	5	0.88	4	0.88	4
dph	0.26	2	1.00	9	1.00	9
echocardiogram	0.44	5	0.50	6	0.50	6
flare	0.93	2	0.02	9	0.02	9
german	0.88	5	0.96	23	1.00	22
glass	0.80	5	0.57	9	0.57	9
hayes-roth	0.69	5	0.62	4	0.73	1
h-d	0.81	5	0.97	9	0.96	10
hea	0.84	5	0.97	12	0.99	9
hea_noise	0.85	5	0.84	19	0.86	16
heart	0.82	5	0.99	12	1.00	9

Datasets name	Manual search method (2–5)		BIC method		Laplace method	
	test set R_{ij}	best degree	test set R_{ij}	estimated optimal degree	test set R_{ij}	estimated optimal degree
hepatitis	0.15	4	0.65	18	0.55	10
horse-23	0.46	3	0.63	21	0.63	21
horse-colic	1.00	2	1.00	26	1.00	26
house-votes-84	0.69	5	0.00	15	0.89	13
ionosphere	0.90	2	0.34	32	0.78	25
iris	0.77	5	0.69	3	0.69	3
khan	0.93	2	0.51	4	0.51	4
labor-neg	1.00	4	0.67	15	0.67	14
lenses	1.00	3	1.00	4	1.00	3
letter-a	0.84	5	1.00	15	1.00	15
lung-cancer	1.00	2	1.00	30	1.00	5
lymphography	0.56	3	0.81	17	0.81	16
mha	0.20	2	0.91	7	0.91	7
monk1	1.00	3	0.89	5	0.98	4
monk2	1.00	3	0.51	5	0.12	4
monk3	0.96	2	0.42	5	0.56	4
mushroom	0.98	5	0.93	10	0.93	10
musk1	0.98	3	0.00	165	0.00	159
musk2	0.45	5	1.00	14	1.00	14
nettalk_stress	0.88	5	0.84	4	0.84	4
new-thyroid	0.24	5	0.21	4	0.21	4
page-blocks	0.97	5	0.96	9	0.96	9
pendigits-8	0.98	4	1.00	14	1.00	14
pha	0.45	2	0.00	8	0.00	8
phm	0.86	5	0.89	10	0.89	10
phn	0.89	2	0.94	8	0.94	8
pid	0.73	2	0.06	5	0.06	5
pid_noise	0.54	5	0.91	14	0.91	14
pima	1.00	2	0.64	6	0.64	6
poh	0.89	5	0.98	10	0.98	10
post-operative	0.20	3	0.24	7	0.24	7
primary-tumor	0.04	4	0.04	16	0.04	16
pro	0.74	3	0.92	11	0.92	11
promoter	0.93	3	0.85	56	0.80	1
pvro	0.15	3	0.00	17	0.00	17
rph	0.58	4	0.89	7	0.89	7
shuttle-landing-control	1.00	2	1.00	4	1.00	4
sick-euthyroid	0.95	5	1.00	24	1.00	24
sma	0.75	5	0.69	6	0.69	6
smo	0.33	5	0.77	7	0.77	7
smo_noise	0.51	4	0.96	14	0.96	12
sonar	1.00	3	0.77	26	0.75	56
splice	0.98	3	1.00	59	0.95	6
switzerland-heart	0.92	2	0.42	7	0.42	7
t_series	0.64	3	0.43	1	0.43	1
tae	0.95	4	0.95	4	0.95	4
tae_noise	0.71	5	0.83	9	0.83	8
thy_noise	1.00	5	0.98	6	0.99	9
tic-tac-toe	0.68	5	0.69	8	0.69	8
titanic	0.69	3	0.00	2	0.00	2
tmris	1.00	2	1.00	2	1.00	2
tqr	0.96	5	0.99	10	0.99	9
trains-transformed	1.00	2	1.00	8	1.00	7
ttt	0.62	5	0.63	8	0.63	8
va-heart	0.36	3	0.00	7	0.00	6
veh	1.00	5	0.32	17	0.32	17
veh_noise	1.00	5	0.29	29	0.29	29
vot_noise	0.99	3	0.67	29	0.67	27
wdbc	0.87	3	0.74	16	0.81	29
wine	1.00	5	1.00	9	1.00	9

Datasets name	Manual search method (2–5)		BIC method		Laplace method	
	test set R_{ij}	best degree	test set R_{ij}	estimated optimal degree	test set R_{ij}	estimated optimal degree
wpbc	0.39	3	0.81	19	0.94	32
xaa	0.83	5	0.44	17	0.44	17
xab	0.85	5	0.09	17	0.09	17
xac	0.93	4	0.38	17	0.38	17
xad	0.89	3	0.24	17	0.24	17
xae	0.73	5	0.00	17	0.00	17
xaf	0.77	3	0.15	17	0.15	17
xag	1.00	2	0.33	17	0.33	17
xah	0.85	2	0.62	17	0.62	17
xai	0.90	4	0.46	17	0.46	17
yha	0.99	5	0.98	8	0.98	8
zoo	1.00	3	0.29	14	0.29	15