Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 70 (2006) 173–186
NEUROCOMPUTING
www.elsevier.com/locate/neucom

A meta-learning approach to automatic kernel selection for support vector machines

Shawkat Ali[a,*], Kate A. Smith-Miles[b]

[a]School of Information Systems, Central Queensland University, Qld 4702, Australia
[b]School of Engineering and Information Technology, Deakin University, VIC 3125, Australia

Received 5 April 2005; received in revised form 28 March 2006; accepted 29 March 2006
Communicated by D.R. Musicant
Available online 30 June 2006

Abstract

Appropriate choice of a kernel is the most important ingredient of the kernel-based learning methods such as support vector machine (SVM). Automatic kernel selection is a key issue given the number of kernels available, and the current trial-and-error nature of selecting the best kernel for a given problem. This paper introduces a new method for automatic kernel selection, with empirical results based on classification. The empirical study has been conducted among five kernels with 112 different classification problems, using the popular kernel based statistical learning algorithm SVM. We evaluate the kernels' performance in terms of accuracy measures. We then focus on answering the question: which kernel is best suited to which type of classification problem? Our meta-learning methodology involves measuring the problem characteristics using classical, distance and distribution-based statistical information. We then combine these measures with the empirical results to present a rule-based method to select the most appropriate kernel for a classification problem. The rules are generated by the decision tree algorithm C5.0 and are evaluated with 10 fold cross validation. All generated rules offer high accuracy ratings.

Keywords: Support vector machine; Kernels; Automatic selection; Classification

1. Introduction

Recently, researchers in the area of pattern recognition have given more attention to kernel-based learning algorithms due to their strong performance in the area of bioinformatics [10], text mining [19], fraud detection [11], speaker identification [32] and database marketing [3], amongst many others. Support vector machines (SVMs) [28,6,5] are one of the most popular kernel-based learning algorithms, first introduced by Vapnik and his group in the mid-1990s. SVM is an optimal hyperplane (OH)-based statistical learning method, which solves classification as well as regression problems. The performance of the SVM method depends however on the suitable selection of a kernel. A kernel is one of the most important features of the SVM algorithm: it generates the dot products in the higher dimensional feature space. The space could theoretically be of infinite dimension, where linear discrimination is possible. Up to now a good number of kernels have been proposed by researchers, but there has been little research conducted to advise how to choose an appropriate kernel for a given problem [22,20]. Clearly, there is a need for automatic kernel selection methods.

Our present research is the first step to provide the solution to this issue. Our aim is to provide an answer to the following question: given the characteristics of a classification data set, and default parameter settings of the SVM,[1] which kernel is likely to produce the most accurate results? Our methodology seeks to understand the characteristics of the data (classification problem),

*Corresponding author. Tel.: +61 7 4930 9880; fax: +61 7 4930 9729.
E-mail addresses: s.ali@cqu.edu.au (S. Ali), Kate.Smith@infotech.monash.edu.au (K.A. Smith-Miles).

[1]The methodology proposed could also be extended to consider experimentation with the SVM parameters, for example, to find rules to describe the best value of the regularisation parameter given the characteristics of the data set.

0925-2312/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2006.03.004

understand which kernels perform well on which types of problems, and generate rules to assist in the automatic selection of kernels for SVMs. This is a meta-learning approach [31]. First, we identify the data set characteristics matrix by statistical measures as we have done in some previous related work [24,23]. All the statistical formulations are available in Matlab statistics toolbox [25]. We then build models for 112 classification problems (see Appendix A) from the UCI Repository [4] and Knowledge Discovery Central [13] database using SVM with five different kernels, and employing a cross-validation testing methodology. Finally, we use the induction algorithm C5.0 (Windows version See 5, http://www.rulequest.com/see5-info.html) to generate the rules to describe which of the five kernels is most suitable for which type of problem, given the data set characteristics and the performance of each kernel on each data set. We also examine the rules by 10 fold cross validation performance.

Our paper is organised as follows: in Section 2, we provide some theoretical frameworks regarding SVM. Section 3 focuses on a theoretical analysis of kernels, and draws a distinction between the five kernels considered in this paper. A comprehensive performance evaluation of all five kernels on the set of 112 classification problems is the presented in Section 4. Section 5 describes the statistical measures and methodology used to generate the data set characteristics matrix. The performance results and the data characteristics are then combined using the rule based learning algorithm C5.0, and the rules for automatic kernel selection are presented and evaluated in Section 6. Finally, we conclude our research in Section 7.

2. Support vector machine

Let us consider a binary classification task, the data matrix $D = (x_1, y_1), \ldots, (x_\ell, y_\ell)$, $x \in R^n$, $y \in \{-1, +1\}$ having corresponding targets $y_1, \ldots, y_\ell$. Our aim is to find the OH in the feature space with this data matrix. The OH separates the classes of the data points without error by maximising the distance between the closest vectors as shown in Fig. 1.

We refer the interested reader to [30,29] for a more comprehensive discussion of SVMs, and their underlying mathematics. For the purposes of this paper, we need only mention that the construction of non-linear kernels for transforming the original data plays a critical role in the SVM's ability to separate the data for classification purposes. Kernel theory is presented in the next section.

3. Kernel theory

Since the development of SVM as an effective classifier for binary class problems, great interest has been generated in the method used to generalise the linear decision rule to non-linear ones, using kernel functions. A kernel function $K(x_i, x_j)$ is a transformation function that satisfies *Mercer's Theorem* [29]. It basically explains that the kernel matrix has to be semi-definite, that means only has positive eigenvalues. Linear methods like principle components analysis (PCA) and Fisher discriminant (FD) analysis have been formulated using kernels, producing the new techniques kernel-PCA (KPCA) and kernel-FD (KFD) [17]. Due to their emphasis on kernels, these methods have become known as kernel machines [17,7]. A simple data set transformation by a kernel is shown in Fig. 2. A linear separation is produced in the feature space.

Kernel maps the patterns x_i to a higher dimensional feature space F, where a linear separation is feasible. The kernel function K could be defined as the inner product of two transformed input vectors as follows:

$$K(x_1, x_2) = \langle \Phi(x_1)\Phi(x_2) \rangle.$$

The linear, polynomial and radial basis function (rbf), are the most commonly used kernels for SVM. We formulate the SVM classical kernels as follows [28,33]:

The linear kernel function is

$$K(x_i, x_j) = \langle x_i^{\mathrm{T}} x_j \rangle.$$

The dth order polynomial kernel function is

$$K(V) = \langle x_i^{\mathrm{T}} x_j \rangle^d$$

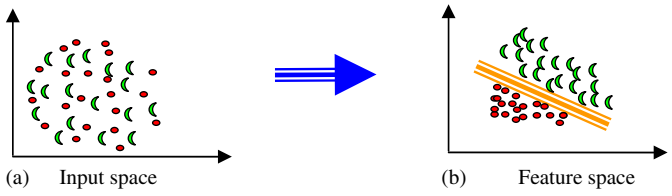

Fig. 2. The linear discrimination boundary on an artificial data set.

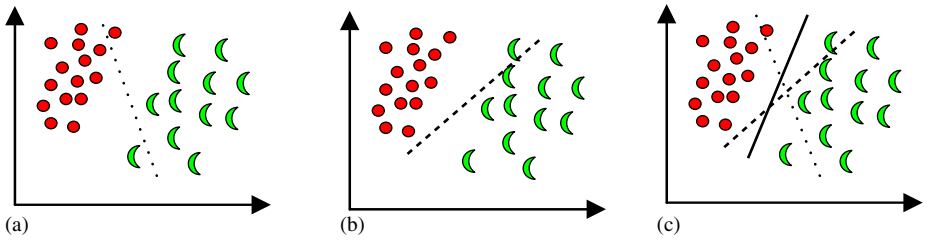

Fig. 1. The training patterns are represented by two different symbols based on their class labels. These patterns are classified by non OH in (a) and (b), but the solid line indicates the OH in (c).

or

$$K(x_i, x_j) = (\langle x_i^{\mathrm{T}} x_j \rangle + 1)^d.$$

Vapnik suggested choosing the second function as the polynomial kernel, which avoids the problems of the hessian matrix becoming zero [5].

For example a second-order polynomial kernel is

$$\begin{aligned}((x_i^{\mathrm{T}} x_j) + 1)^2 &= (x_{i1}x_{j1} + x_{i2}x_{j2})^2 + 2(x_{i1}x_{j1} + x_{i2}x_{j2}) + 1 \\ &= (x_{i1}x_{j1})^2 + 2(x_{i1}x_{j1})(x_{i2}x_{j2}) + (x_{i2}x_{j2})^2 \\ &\quad + 2(x_{i1}x_{j1}) + 2(x_{i2}x_{j2}) + 1.\end{aligned}$$

Finally, the polynomial transformation is

$$K(x_i, x_j) = (1, x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}).$$

rbf has received significant attention in SVM implementations. The classical function of this kernel is

$$K(x_i, x_j) = \exp\{-|x_i - x_j|^2\}.$$

Boser, Guyon, and Vapnik [29,18,15] modified the classical function by introducing a smoothing parameter h as follows:

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2h^2}\right),$$

where $h > 0$.

The finite spline kernel [9] can be described as follows:

$$\begin{aligned}K(x_i, x_j) = 1 + (x_i^{\mathrm{T}} x_j) + \frac{1}{2}(x_i^{\mathrm{T}} x_j)\min(x_i^{\mathrm{T}} x_j)^2. \\ - \frac{1}{6}\min(x_i^{\mathrm{T}} x_j)^3.\end{aligned}$$

Another positive semi definite kernel is called multiquadratic [8]:

$$K(x_i, x_j) = (\|x_i - x_j\|^2 + \tau^2)^{1/2},$$

where $\tau > 0$.

In our previous work [1], we have formulated a new kernel, called Laplace, for the SVM family. The double exponential or Laplace distribution [16] for a random variable x is as follows:

$$f_X(x; \alpha, \beta) = \frac{1}{2\beta}\exp\left(-\frac{|x - \alpha|}{\beta}\right),$$

where $-\infty < \alpha < \infty$ and $\beta > 0$.

Therefore, we can present the Laplace kernel [1] with smoothing parameter h as

$$K(x_i, x_j) = \exp\left(-\frac{|x_i - x_j|}{h}\right).$$

A graphical comparison among all kernels is explained in Fig. 3 for a binary class synthetic problem. The rectangular and the cross signs indicate the two different classes of the problem. The middle lines perform the OH function for the classification.

The rbf and Laplace kernels are the only ones to classify all patterns perfectly on this artificial data set. It is interesting to observe from these figures how each kernel generates the OH, and how certain kernels are inherently quite limited in their ability to find the OH for highly non-separable data.

4. A performance evaluation of Kernal methods

To investigate the effectiveness of the kernels we conducted a wide range of experiments on binary as well as multiclass data sets as shown in Appendix A. We examined 112 classification problems from the UCI Repository [4] and Knowledge Discovery Central [13] database. We use 10 fold cross validation [27], for those data sets with less than 1000 samples (68% of the data sets). Otherwise we use the hold-out method, with 70% of the data randomly extracted for training and the remainder reserved for testing. We report the combined (10 fold cross validation and hold-out) test set results with the best parameter performance for all parametric kernels, after experimenting with a range of parameter values. We observed no significant impact of various values of the regularization parameter C in our SVM implementation, so it was fixed as 'infinite' over the experiments according to the default MATLAB implementation.[2] The smoothing parameter (h) for Laplace kernel is estimated by Parzen algorithm (please see [1] for details). We have summarised the average classification accuracy and standard deviation for the test set with different kernels and a range of parameter values in Appendix B.

4.1. Kernel performance: accuracy

The percentage of classification performance on the test set, averaged across the 112 problems, for the different SVM kernels is shown in Fig. 4.

The rbf and Laplace kernels show very similar performance averaged over all problems. Polynomial kernel was the second most accurate in our experiments, but was clearly less accurate than rbf or Laplace. The third tier of performance was the spline kernel while the multiquadratic kernel performed worst over the experiments.

4.2. Significance tests

The t-test results are summarised in Table 1. We considered the base kernel as polynomial. The test input was the percentage of correct classification for all the

[2]It should be noted that the optimal choice for the regularisation parameter C is contentious (see svms.org/parameters for a discussion). While some researchers believe that the default value of $C = \infty$ (essentially a hard-margin classifier) is appropriate for high-dimensional data sets, others suggest searching for the optimal C value for different data sets. In any case, whether we are using the best C value for each kernel is not so important to the aims of this paper. Our focus is to determine which kernel is expected to perform best given the characteristics of the data set, and other user-defined parameter choices (including C).

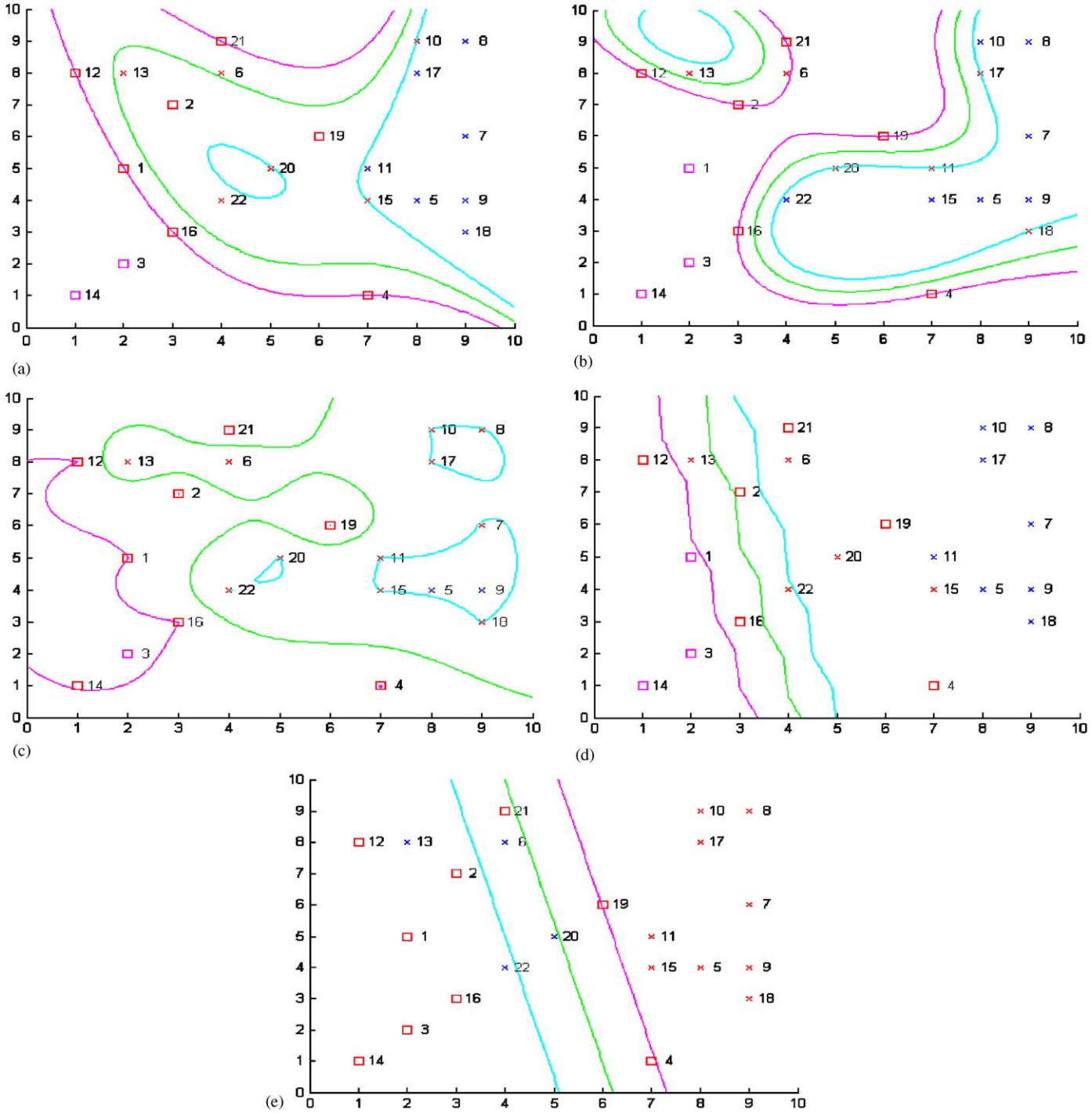

Fig. 3. Graphical representations of the polynomial, rbf, Laplace, spline and multiquadratic kernel on an artificial data set. The cross and rectangular signs indicate the two classes of data. The middle lines of the above graphs represent the OH for classification. Those data points placed on the hyperplane are called SVs: (a) polynomial kernel—3 classification errors, (b) rbf kernel—0 classification errors, (c) Laplace kernel—0 classification errors, (d) spline kernel—5 classification error, (e) multiquadratic kernel—6 classification error.

kernels. The t-test hypothesis is

$$\mathbf{H}_0 : \mu_1 - \mu_2 = 0$$

versus the alternative hypothesis

$$\mathbf{H}_1 : \mu_1 - \mu_2 \neq 0.$$

Comparing the polynomial with Laplace kernel showed no significance performance difference. The higher values of the significance level P suggested accepting the null hypothesis. The 95% CI for these kernels are balance skewed as shown in Table 1. But, comparing the polynomial kernel with rbf, spline and multiquadratic

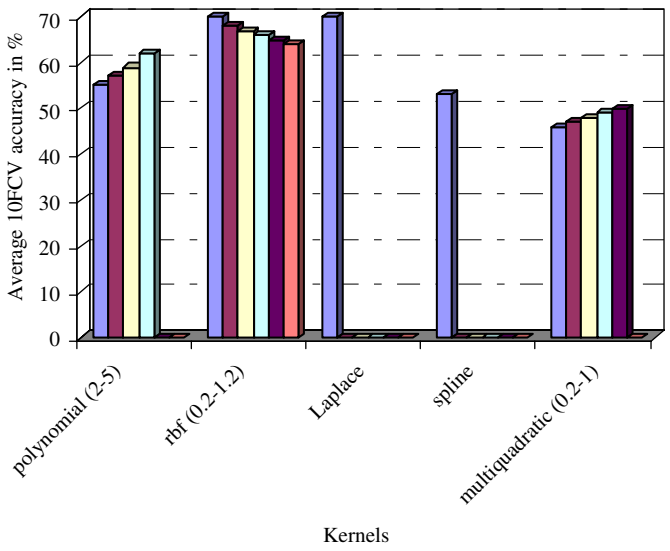

Fig. 4. Comparison of the average classification performance (test set) for different kernels.

Table 1
Results of the *t*-test for all kernels

Algorithms	Hypothesis (**H**)	Significance (*P*)	Confidence interval (CI)	
Polynomial vs. rbf	$\mathbf{H_1}$	0.0343	−12.5794	−0.487
Polynomial vs. Laplace	$\mathbf{H_0}$	0.0764	−11.7133	0.5944
Polynomial vs. spline	$\mathbf{H_1}$	0.0062	2.5033	14.985
Polynomial vs. multiquadratic	$\mathbf{H_1}$	0.00003	5.4435	18.2282

kernels showed there have significant performance difference. The smaller values of the significance level suggested rejecting the null hypothesis. The 95% confidence intervals for these kernels are either highly positively or negatively skewed as shown in Table 1.

It should be noted that, while the focus of our study is on the accuracy of the different kernels, the computational expense of each the kernels is very similar, and the difference is certainly not statistically significant.

It is clear from these results that some kernels are better than others, when considering the average performance across the 112 data sets. It is also clear from inspecting the detailed performance results in Appendix C, that certain kernels are better suited to certain types of problems. For example, while the multiquadratic kernel performs worst on average across the 112 data sets, it performs well on data sets such as allbp, ann1, ann2, breast cancer, dph, flare, letter-a and musk2. Understanding what these data sets have in common and why a certain kernel performs well on them is the focus of the remainder of this paper.

In the following sections, we describe the methodology we use to assist in the appropriate selection of a kernel for a given data set. First each data set is described by a set of measurable characteristics; we then combine this information with the performance results; and finally use a rule-based induction method to provide rules describing when a certain kernel is likely to perform well.

5. Data sets characteristics measurement

Each data set can be described by simple, distance and distribution-based statistical measures [24,23]. Let $X^i_{k,j}$ be the value of the jth variable (column) in the kth example (row) of data set i. These three types of measures characterise the data set matrix in different ways. Firstly, the simple classical statistical measures identify the data characteristics based on variable to variable comparisons (i.e. comparisons between columns of the data set). Then, the distance based measures identify the data characteristics based on sample to sample comparisons (between rows of the data set)). Finally, the density-based measures consider the relationships between single data points and the statistical properties of the entire data matrix to identify the data sets characteristics. All of these statistical measures have been briefly summarised in Appendix D. The simple statistical measures are calculated within each column, and then averaged over all columns to obtain global measures of the data set. Likewise, the distance measures are averaged over all pair wise comparisons, and the density based measures are averaged across the entire matrix.

For each data set i, a total of 29 measures are calculated (11 statistical, 3 distance based, 15 density based). The data set characteristics matrix is then assembled with the columns comprising the 29 measures, and the rows comprising the 112 data sets.

Finally, by combining the data set characteristics with the experimental results presented in Appendix C we can generate rules for automatic kernel selection as described in the following section.

6. Rule generation

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 more interest in the machine learning community. [21] 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 an 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' [21,7]. 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, m should contain at least two branches. For detail formulations see [21].

6.1. Rules for automatic kernel selection

The trial-and-error approach is a very common procedure to select the best kernel. It is an extensive and computationally complex task to find the best kernel by following this procedure. If we are interested in applying the best kernel to a particular problem we have to consider which kernel is more suitable for which problem. The suitability can be assessed from rules generated by the following data-dependant method.

Now that the characteristics of each data set can be quantitatively measured, we can combine this information with the empirical evaluation of kernel performance and construct the data set characteristics matrix. Thus, the result of the jth kernel on the ith data set is calculated as

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

where e_{ij} is the percentage of correct classification for the jth algorithm on data set i, and e_i is a vector of accuracy for data set i. A detailed kernel classification performance has been summarised in Appendix C. We reported the best parameter performance only. The class labels in the matrix are assigned based on the single best performing kernel name. For example, if polynomial kernel shows the best performance for the data set A, then the class label in the matrix for problem A is polynomial. Based on the 112 classification problems we can then train a rule-based classifier (C5.0) to learn the relationship between data set characteristics and the name of the best performing kernel (five possible output classes corresponding to each kernel name). We split the matrix randomly into a 90% training set to construct the model tree for appropriate kernel selection and the remaining 10% of the data is used to evaluate the model. 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 kernel 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 2–5. We found the suitable parameter value for global pruning factor; c is between 60% and 95% and the number of minimum classes, m is 2.

6.1.1. Rules for polynomial kernel

The rules for polynomial kernel are generated with $c = 70\%$ and $m = 2$. Table 2 shows the accuracy of the 10FCV

Table 2
10FCV test set classification accuracy results for the polynomial kernel selection rule

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Average
81.9	80	75.5	82.5	78.1	82.1	81	81.3	76.1	81.9	80.04

Table 3
10FCV test set classification accuracy results for the rbf kernel selection rule

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Average
80	91.7	83.3	91.6	90.1	90.9	91	88.1	80.5	85.5	87.27

Table 4
10FCV test set classification accuracy results for the Laplace kernel selection rule

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Average
90.9	92.9	89.9	94	92.1	94.9	90	91.8	89.9	91.8	91.82

Table 5
10FCV test set classification accuracy results for the spline kernel selection rule

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Average
93.5	90.8	94.6	93.9	94.7	95.8	96.7	95.5	96.6	93.3	94.54

test sets, and the best rule is as follows:

IF md $\Leftarrow$ 218.276, THEN we should choose polynomial kernel for SVM classification.

We observed from the wide group of data sets, polynomial kernel worked better for discrete and nominal data. Moreover, these data sets are not normally distributed, based on examination of histograms. It appears therefore that the polynomial kernel is best suited to non-normally distributed data sets containing discrete and nominal data which satisfy the condition $md <= 218.276$.

6.1.2. Rules for rbf kernel

The rules for rbf kernel are generated with $c = 70\%$ and $m = 2$. Table 3 shows the accuracy of the 10FCV test sets, and the best rule is as follows:

IF ($R > 9$ AND $p_{norm\_cdf} > 7.2957$) OR ($p_{du\_cdf} \Leftarrow 2.8185$) THEN we should choose rbf kernel for SVM classification.

The rbf kernel is suitable for continuous data sets; those that are normally or closely normally distributed. It showed worst performance for categorical data sets. We generated two meta rules satisfied this group of data sets.

6.1.3. Rules for Laplace kernel

The rules for Laplace kernel are generated with $c = 70\%$ and $m = 2$. Table 4 shows the accuracy of the 10FCV test sets, and the best rule is as follows:

IF $y_{gama\_pdf} \Leftarrow 17.1671$ THEN we should choose Laplace kernel for SVM classification.

The Laplace kernel showed similar characteristics with rbf kernel. It is also worked better for a combination of discrete and continuous data sets. But the out come of this combination should be close to normally distributed.

6.1.4. Rules for spline kernel

The rules for spline kernel are generated with $c = 70\%$ and $m = 2$. Table 5 shows the accuracy of the 10FCV test sets, and the best rule is as follows:

IF ($y_{ray\_pdf} > 20.2875$) OR ($M \Leftarrow 90.7233$) THEN we should choose spline kernel for SVM classification.

The spline kernel is suitable for nominal and categorical data sets. It showed worst performance for continuous data sets.

The generated rules showed more than 80% accuracy in our method. These rules might be useful to determine which kernel is most appropriate for which problem. The approach taken here produces guidelines (rules) that are quite "black-box" in nature, and an ability to provide greater insight into why certain kernels are suited to different data sets is limited by the multivariate nature of the analysis. Nevertheless, the high accuracy of the generated rules, and the cross-validation methodology adopted, ensures the usefulness of the rules to

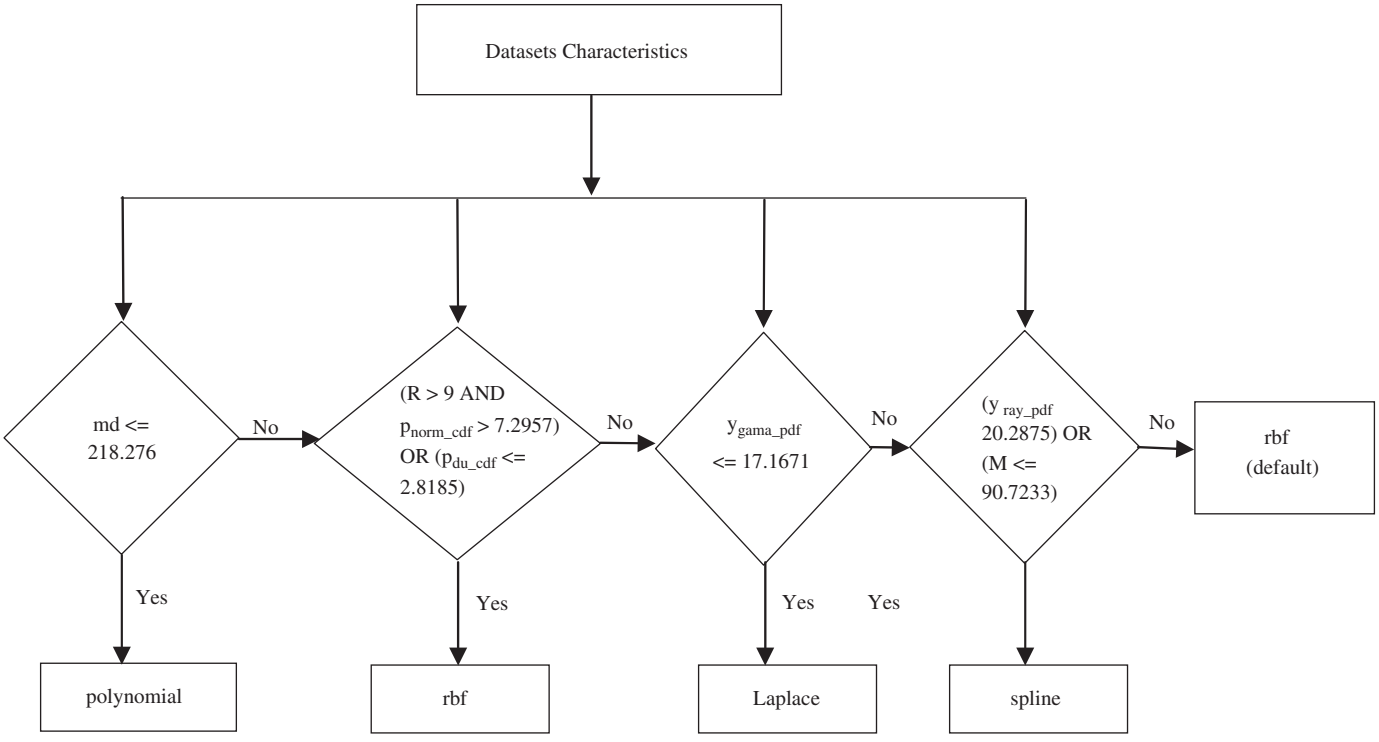

Fig. 5. A flowchart for automatic kernel selection.

assist in kernel selection. We cannot generate the rules for multiquadratic kernel due to insufficient data sets for which this kernel performed best. We summarise the above rules for automatic kernel selection in a flowchart as shown in Fig. 5.

The rbf kernel is setup as a default kernel in our automatic kernel selection method due to its best average performance over the experiments.

7. Conclusions

In this paper, we have presented a new rule based method for automatic kernel selection based on statistical measures of the data sets and extensive empirical performance results. This method is very simple, efficient and allows the computational complexity for selection of kernels to be reduced. Empirical results on a wide range of problems point out that all rules are acceptable due to their higher accuracy performance. Most of the descriptive statistical measures are more appropriate for univariate data analysis. So it is difficult to explain the rules and gain insight based on statistical meta attributes for multivariate data. Therefore, we generated the rules based on meta attributes information and select to explain the kernel behaviour based on data distribution and types. We recommend rbf be used as the default kernel in our rule based kernel selection method, due to its best performance when averaged across all the data sets. If a data set satisfies the conditions of any of our rules however, a more appropriate kernel can be chosen. Considering the performance of the above rules, we have been found C5.0 to be a very good rule generation tool for automatic kernel selection.

This research has opened the way for a number of significant research directions. Firstly, the meta-learning methodology described in this paper can be extended to assist in the automatic selection of kernel parameters. SVM parameters, such as the regularisation parameter C, could be treated as variables in the methodology. Results could be generated for each kernel (or even a fixed kernel choice) and the decision tree could be used to find rules that explain which values of C are suited to certain data sets. Secondly, a similar methodology could be used for the automatic selection of learning algorithms. We have already started to investigate this direction for comparing the suitability of different learning algorithms on different data sets, including neural networks, SVMs, and other machine learning methods [2]. Thirdly, now that the matrix of data set characteristics and performance results has been generated, we can consider if other rule generation methods can produce better rules. We can also explore alternative approaches such as clustering [24] in an attempt to gain greater insight into why certain techniques perform better on certain types of problems.

Acknowledgements

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

Appendix A. Data set descriptions

A wide range of experiments on binary as well as multiclass data sets are shown in Table A1.

Appendix B. Kernel performance for the 10FCV test data (% of accuracy)

The average classification accuracy and standard deviation for the test set with different kernels and a range of parameter values are shown in Table B1.

Appendix C. Kernel performance with best parameter

Combination of the data set characteristics with the experimental results is presented in Table C1.

Appendix D

D.1. Simple statistical measures

Descriptive statistics can be used to summarise any large data set into a few numbers that contain most of the relevant characteristics of that data set. The following section lists some of the measures provided by the Matlab Statistics Toolbox [25] and some other different sources [14,26,12] as follows:

(1) *Geometric mean* (*G_mean*): The geometric mean of a sequence $\{X_i\}_{i=1}^n$ is

$$G\_\text{mean} = \left[\prod_{i=1}^{n} X_i\right]^{1/n}.$$

(2) *Harmonic mean* (*H_mean*): The harmonic mean $H(X_1, \ldots, X_n)$ of n points X_i is

$$H\_\text{mean} = \frac{n}{\sum_{i=1}^{n}(1/X_i)}.$$

(3) *Trim mean* (*T_mean*): The trim mean measures the arithmetic mean of a sample X excluding the highest and lowest specified trim fraction of the observations. The trim fraction is a user dependant parameter. We consider 20 for this parameter value over the experiment. The trimmed mean is a robust estimate of the centre location of a sample. For data sets containing outliers, the trimmed mean is a more appropriate estimation of the centre of the data set.

(4) *Standard deviation* (*std*): The standard deviation (std) measures the spread of a set of data as a proportion of its mean:

$$std = \left(\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^{1/2},$$

where

$\bar{x} = \frac{1}{n}\sum x_i$, n is the sequence of length.

Table A1

# data sets	Data set names	# samples	# attributes	# classes	# data sets	Data set names	# samples	# attributes	# classes
1	abalone	1253	8	3	57	Musk2	1154	15	2
2	adp	1351	11	3	58	nettalk_stress	1141	7	5
3	adult-stretch	20	4	2	59	new-thyroid	215	5	3
4	allbp	840	6	3	60	Page-blocks	1149	10	5
5	ann1	1131	6	3	61	pendigits-8	1399	16	2
6	ann2	1028	6	3	62	Pha	1070	9	5
7	aph	909	18	2	63	Phm	1351	11	3
8	art	1051	12	2	64	Phn	1500	9	2
9	australian	690	14	2	65	Pid	532	7	2
10	balance-scale	625	4	3	66	pid_noise	532	15	2
11	bcw	699	9	2	67	Pima	768	8	2
12	bcw_noise	683	18	2	68	Poh	527	11	2
13	bld	345	6	2	69	post-operative	90	8	3
14	bld_noise	345	15	2	70	primary-tumor	339	17	2
15	bos	910	13	3	71	Pro	1257	12	2
16	bos_noise	910	25	3	72	promoter	106	57	2
17	breast-cancer	286	6	2	73	Pvro	590	18	2
18	breast-cancer-wisconsin	699	9	2	74	Rph	1093	8	2
19	bupa	345	6	2	75	shuttle	1450	9	5
20	c	1500	15	2	76	shuttle-landing-control	15	6	2
21	cleveland-heart	303	13	5	77	sick-euthyroid	1582	15	2
22	cmc	1473	9	3	78	Sma	409	7	4
23	cmc_noise	1473	15	3	79	Smo	1429	8	3
24	crx	490	15	2	80	smo_noise	1299	15	3
25	dar	1378	9	5	81	Sonar	208	60	2
26	dhp	1500	7	2	82	splice	1589	60	3
27	dna	2000	60	3	83	switzerland-heart	123	8	5
28	dna_noise	2000	80	3	84	t_series	62	2	2
29	DNA-n	1275	60	3	85	Tae	151	5	3
30	dph	590	10	2	86	tae_noise	151	10	3
31	echocardiogram	131	7	2	87	Thy	1887	21	3
32	flare	1389	10	2	88	thy_noise	3772	35	3
33	german	1000	24	2	89	tic-tac-toe	958	9	2
34	glass	214	10	6	90	titanic	2201	3	2
35	hayes-roth	160	5	3	91	Tmris	100	3	2
36	h-d	303	13	2	92	Tqr	1107	11	2
37	hea	270	13	2	93	trains-transformed	10	16	2
38	hea_noise	270	20	2	94	Ttt	958	9	2
39	heart	270	13	2	95	va-heart	200	8	4
40	hepatitis	155	19	2	96	Veh	846	18	4
41	horse-23	368	22	2	97	veh_noise	761	30	4
42	house-votes-84	435	16	2	98	vot_noise	391	30	2
43	hypothyroid	1265	25	2	99	wdbc	569	30	2
44	ionosphere	351	33	2	100	Wine	178	13	3
45	iris	150	4	3	101	wpbc	199	33	2
46	khan	1063	5	2	102	Xaa	94	18	4
47	labor-neg	40	16	2	103	Xab	94	18	4
48	lenses	24	5	3	104	Xac	94	18	4
49	letter-a	1334	16	2	105	Xad	94	18	4
50	lymphography	148	18	8	106	Xae	94	18	4
51	mha	1269	8	4	107	Xaf	94	18	4
52	monk1	556	6	2	108	Xag	94	18	4
53	monk2	601	6	2	109	Xah	94	18	4
54	monk3	554	6	2	110	Xai	94	18	4
55	mushroom	1137	11	2	111	Yha	1601	9	2
56	musk1	476	166	2	112	Zoo	101	16	7

Some other simple statistical measures including mean, mad, variance, and prctiles, for details see [25].

(5) *Range* (R): The range is the difference between the maximum and minimum values with in a variable.

(6) *Median* (M): The median is the value halfway through the sorted data set, below and above which there lies an equal number of data values.

(7) *Interquartile range* (IQR): The IQR is used as a robust measure of scale and measures the distance between

Table B1

Kernel	Polynomial				rbf						Laplace
Parameter	2	3	4	5	0.2	0.4	0.6	0.8	1.0	1.20	Parzen
Mean	55.62	58.22	59.34	63.25	69.78	69.77	68.37	66.74	65.74	64.90	68.81
Standard Deviation	24.43	25.29	23.73	23.59	22.31	22.79	23.71	24.36	24.86	25.01	23.14

Kernel	spline	Multiquadratic				
Parameter		0.2	0.4	0.6	0.8	1
Mean	54.51	49.35	49.62	50.03	50.63	51.41
Standard Deviation	23.80	25.78	25.61	25.47	25.25	24.93

Table C1

Data sets name	Polynomial		rbf		Laplace	Spline	Multiquadratic	
	Accuracy	Best parameter	Accuracy	Best parameter	Accuracy	Accuracy	Accuracy	Best parameter
abalone	0.99	2	0.96	0.4	0.94	1.00	0.01	0.2
adp	0.86	2	1.00	1.2	0.98	0.54	0.54	0.2
adult-stretch	1.00	2	1.00	0.2	1.00	0.00	0.00	0.2
allbp	0.96	5	0.97	0.6	0.94	1.00	1.00	0.2
ann1	0.52	4	1.00	0.4	0.97	0.99	0.99	0.2
ann2	0.03	5	0.80	0.2	0.80	1.00	1.00	0.2
aph	1.00	3	0.96	0.2	0.96	0.92	0.92	0.2
art	1.00	4	1.00	0.2	1.00	0.00	0.00	0.2
australian	0.74	5	1.00	0.4	0.95	0.42	0.08	1
balance-scale	1.00	2	0.91	1.2	0.84	0.88	0.01	1
bcw	0.97	3	1.00	0.2	0.99	0.93	0.00	0.8
bcw_noise	0.95	4	0.99	0.2	1.00	0.47	0.00	1
bld	0.84	2	1.00	0.6	0.81	0.00	0.53	0.2
bld_noise	0.69	2	0.83	0.8	0.00	1.00	0.89	0.2
bos	0.82	5	1.00	0.2	1.00	0.39	0.01	1
bos_noise	0.92	5	0.99	0.6	1.00	0.69	0.01	0.8
breast-cancer	0.57	2	0.42	0.2	0.00	1.00	0.96	0.2
breast-cancer-wisconsin	0.97	4	1.00	0.2	0.99	0.91	0.01	0.2
bupa	0.74	2	1.00	0.6	0.81	0.00	0.64	0.2
c	1.00	2	0.93	1	0.83	0.02	0.02	0.6
cleveland-heart	0.43	5	0.72	0.6	0.69	0.94	0.87	0.2
cmc	0.90	2	0.35	0.2	0.29	0.97	0.71	0.2
cmc_noise	0.75	5	0.86	0.6	0.78	1.00	0.81	0.2
crx	0.67	5	1.00	0.4	1.00	0.00	0.16	1
dar	1.00	5	0.91	0.6	0.91	0.71	0.71	0.2
dhp	0.92	5	1.00	1	0.97	0.51	0.51	0.8
dna	1.00	5	0.67	0.2	0.42	0.00	0.54	0.2
dna_noise	1.00	5	0.33	0.4	0.08	0.00	0.22	0.2
DNA-n	1.00	5	0.79	0.4	0.55	0.00	0.62	0.2
dph	0.29	2	0.86	0.2	0.84	0.94	1.00	0.2
echocardiogram	0.44	5	1.00	0.4	0.97	0.97	0.78	0.2
flare	0.93	2	0.09	0.2	0.15	0.90	1.00	0.2
german	0.85	5	0.85	1.2	0.62	0.76	0.85	0.2
glass	0.80	5	1.00	0.4	0.91	0.04	0.08	0.8
hayes-roth	0.71	5	1.00	0.2	0.30	0.75	0.12	1
h-d	0.81	5	1.00	0.6	0.96	0.96	0.06	1
hea	0.84	5	1.00	0.6	0.94	1.00	0.09	1
hea_noise	0.85	5	1.00	0.6	0.93	0.85	0.08	0.8
heart	0.85	5	0.96	0.6	0.94	1.00	0.02	1
hepatitis	0.15	4	1.00	0.6	0.65	0.60	0.30	0.2
horse-23	0.46	3	0.51	0.2	0.51	1.00	0.63	0.2
house-votes-84	0.38	5	1.00	1.2	0.72	0.68	0.00	0.2
hypothyroid	0.95	4	0.76	0.2	0.52	1.00	0.05	0.4
ionosphere	0.90	2	1.00	0.2	0.96	0.27	0.27	1
iris	0.77	5	1.00	1.2	0.98	0.02	0.32	1

Table C1 (*continued*)

Data sets name	Polynomial		rbf		Laplace	Spline	Multiquadratic	
	Accuracy	Best parameter	Accuracy	Best parameter	Accuracy	Accuracy	Accuracy	Best parameter
khan	0.93	2	0.94	0.2	0.93	1.00	0.00	0.8
labor-neg	1.00	4	0.00	0.2	0.00	0.67	0.67	0.2
lenses	1.00	3	1.00	0.2	1.00	0.44	0.44	0.2
letter-a	0.84	5	0.99	0.2	0.98	0.00	1.00	0.2
lymphography	0.61	3	0.61	0.2	0.61	0.94	0.94	0.2
mha	0.34	2	1.00	1	1.00	0.64	0.64	0.2
monk1	1.00	3	0.92	0.6	0.92	0.08	0.01	1
monk2	1.00	3	0.91	0.2	0.89	0.04	0.31	0.2
monk3	0.96	2	0.58	0.2	0.58	1.00	0.05	1
mushroom	0.98	5	1.00	0.2	1.00	0.69	0.00	0.2
musk1	0.99	3	1.00	0.2	0.00	0.29	0.29	0.2
musk2	0.47	5	1.00	0.2	0.99	0.00	1.00	0.2
nettalk_stress	0.88	5	1.00	0.6	0.96	0.75	0.29	1
new-thyroid	0.48	5	1.00	1.2	0.88	0.55	0.55	0.2
page-blocks	0.97	5	1.00	0.8	0.94	0.98	0.98	0.2
pendigits-8	0.99	4	1.00	0.2	1.00	0.56	0.56	0.2
pha	0.36	2	0.97	0.4	1.00	0.86	0.86	0.2
phm	0.88	5	1.00	0.6	0.99	0.79	0.79	0.2
phn	0.89	2	1.00	0.2	1.00	0.51	0.36	1
pid	0.58	2	0.81	0.4	0.77	0.00	0.12	0.2
pid_noise	0.46	5	0.85	0.4	0.24	0.04	0.17	0.2
pima	1.00	2	0.99	0.4	0.96	0.00	0.81	0.2
poh	0.89	5	1.00	1	1.00	0.00	0.95	0.2
post-operative	0.41	3	0.35	0.4	0.38	0.97	0.97	0.2
primary-tumor	0.04	4	0.18	1	0.21	0.98	0.91	0.2
pro	0.79	3	1.00	1	0.97	0.76	0.76	0.2
promoter	0.97	3	1.00	0.2	1.00	0.00	0.07	0.4
pvro	0.14	3	0.94	0.2	0.94	0.10	1.00	0.2
rph	0.58	4	1.00	0.8	0.99	0.47	0.47	1
shuttle	1.00	4	0.98	1.2	0.63	0.96	0.96	0.2
shuttle-landing-control	0.41	3	0.35	0.4	0.38	0.97	0.97	0.2
sick-euthyroid	0.99	5	1.00	1.2	0.97	0.99	0.99	0.2
sma	0.75	5	1.00	0.2	1.00	0.46	0.46	0.2
smo	0.33	5	0.26	0.2	0.27	0.31	1.00	0.2
smo_noise	0.51	4	0.60	0.2	0.46	0.02	0.99	0.2
sonar	1.00	3	0.99	0.2	0.99	0.18	0.18	0.2
splice	1.00	3	0.69	0.2	0.31	0.27	0.39	0.2
switzerland-heart	0.95	2	0.74	0.2	0.71	1.00	1.00	0.2
t_series	0.64	3	1.00	1.2	0.93	1.00	0.57	0.6
tae	0.95	4	0.76	0.2	0.52	1.00	0.05	0.4
tae_noise	0.71	5	1.00	0.4	0.95	0.54	0.07	0.8
thy	1.00	5	0.83	1.2	0.75	0.99	0.99	0.2
thy_noise	1.00	5	0.83	1.2	0.75	0.99	0.99	0.2
tic-tac-toe	0.69	5	0.97	0.2	1.00	0.44	0.66	0.2
titanic	0.69	3	1.00	0.2	1.00	0.99	0.00	0.2
tmris	1.00	2	0.99	0.2	0.99	0.12	0.01	0.6
tqr	0.96	5	1.00	0.4	1.00	0.00	0.06	1
trains-transformed	0.33	5	0.26	0.2	0.27	0.31	1.00	0.2
ttt	0.63	5	1.00	0.6	0.94	0.40	0.60	0.2
va-heart	0.26	3	0.53	1.2	0.00	0.11	0.53	0.2
veh	1.00	5	0.79	0.6	0.72	0.00	0.55	1
veh_noise	1.00	5	0.74	0.8	0.53	0.00	0.54	1
vot_noise	0.97	3	0.98	1.2	0.73	0.49	0.00	0.2
wdbc	0.87	3	1.00	0.4	0.92	0.38	0.24	1
wine	1.00	5	0.99	0.2	1.00	0.29	0.11	1
wpbc	0.39	3	0.72	0.2	0.72	1.00	1.00	0.2
xaa	0.83	5	1.00	0.8	1.00	0.24	0.07	1
xab	0.85	5	1.00	0.4	0.87	0.17	0.04	0.2
xac	0.93	4	1.00	1.2	0.93	0.33	0.04	0.4
xad	0.89	3	1.00	0.8	0.78	0.16	0.08	1
xae	0.81	5	1.00	0.4	0.95	0.05	0.05	0.8
xaf	0.77	3	1.00	0.6	0.77	0.08	0.08	0.4

Table C1 (*continued*)

Data sets name	Polynomial		rbf		Laplace	Spline	Multiquadratic	
	Accuracy	Best parameter	Accuracy	Best parameter	Accuracy	Accuracy	Accuracy	Best parameter
xag	1.00	2	0.85	1.2	0.67	0.00	0.08	0.8
xah	0.85	2	1.00	1.2	0.77	0.25	0.02	0.6
xai	0.90	4	1.00	0.2	0.95	0.36	0.08	1
yha	1.00	5	1.00	0.2	1.00	0.54	0.54	0.2
zoo	1.00	3	1.00	0.2	1.00	0.29	0.00	0.2

the 25th and the 75th percentile [14]. The hypothesis is, if the variables are approximately normal, then $IQR/\sigma \approx 1.3$, where σ is the standard deviation of the population.

(8) *Max. and Min. eigenvalue*: The maximum and minimum eigenvalues is the maximum and minimum variance of a data set. We use the sample covariance matrix to calculate the eigenvalues:

$$\Re = \frac{1}{n}\sum_{p_{i=1}}^{n} X(p_i)' X(p_i).$$

(9) *Skewness and kurtosis*: The skewness (s) and kurtosis (k) [25] of a distribution is defined as

$$s = \frac{\mathbf{E}(X-\mu)^3}{\sigma^3}, \quad k = \frac{\mathbf{E}(X-\mu)^4}{\sigma^4},$$

where μ is the mean and σ the standard deviation of X.

(10) *Correlation coefficient*: The sample correlation coefficient between X and Y is denoted by r_{xy} or simply by r [26] as follows:

$$r = \frac{s_{xy}}{s_x s_y} = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{X_i - \bar{X}}{\sigma_x}\right)\left(\frac{Y_i - \bar{Y}}{\sigma_y}\right).$$

(11) *Z-score*: The value of Z-score is greater than 3 indicates that the data distribution has outliers [26]:

$$Z\_\text{score} = \frac{X - \bar{X}}{\sigma}.$$

D.2. Distance-based measures

The distance-based measures calculate the dissimilarity between samples. A list of different distance-based measured is mentioned as followed [25]:

(1) *Euclidean distance* (*ed*): The Euclidean distance simply calculates the geometric distance of the samples in the multivariate data space as follows:

$$ed_{\text{rs}}^2 = (x_\text{r} - x_\text{s})V^{-1}(x_\text{r} - x_\text{s})',$$

where V is the diagonal matrix.

(2) *Mahalanobis distance* (*md*): The mahalanobis distance calculates the geometric distance of the samples like Euclidean but by constructing the covariance matrix rather than diagonal matrix as follows:

$$md_{\text{rs}}^2 = (x_\text{r} - x_\text{s})V^{-1}(x_\text{r} - x_\text{s})',$$

where V is the covariance matrix.

(3) *City block distance* (*cbd*): The city block distance calculates the sum of the absolute differences between the values of the samples in the multivariate data space as follows:

$$cbd_{\text{rs}} = \sum_{j=1}^{n}\left|x_{\text{r}j} - x_{\text{s}j}\right|.$$

D.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 assume. We measure the probability density function (pdf) and cumulative distribution function (cdf) for all data sets by considering different types of distributions. We have summarised the entire description in this subsection following [25,12]. A list of different pdf and cdf functions are listed in below.

Let us consider a random variable X as a function from a sample space Ω with the real numbers $\Re$ where $X : \Omega \to \Re : \omega_k \mapsto x_k$ with $\omega_k \in \Omega$ and x_k a realisation of the variable X. The cdf could be defined by $F(x) : \Re \to [0, 1]$ as follows:

$$F(x) = P(X \leqslant x), \quad \forall x \in \Re.$$

By considering the discrete random variable X and its cdf $F(x)$, we can define pdf for X as follows:

$$f(x) = P(X = x), \quad \forall x \in \Re$$

and the pdf for a continuous random variable X is as follows:

$$F(x) = \int_{-\infty}^{x} p(t)\,\mathrm{d}t,$$

where $x \in \Re$. For any $m<n$ the probability that X falls within the interval (m, n) is the area under the pdf curve over the interval is

$$P(m<X<n) = \int_{m}^{n} p(x)\,\mathrm{d}x.$$

(1) χ^2 *pdf*: For each element of X, $y_{\text{chis\_pdf}}$ compute the pdf at X of the χ^2 distribution with v degrees of freedom as follows:

$$y_{\text{chis\_pdf}} = f(x|v) = \frac{x^{(v-2)/2}e^{-x/2}}{2^{v/2}\Gamma(v/2)}.$$

(2) χ^2 *cdf*: For each element of X, $p_{\text{chis\_cdf}}$ compute the cdf at X of the χ^2 distribution with v degrees of freedom as follows:

$$p_{\text{chis\_cdf}} = F(x|v) = \int_0^x \frac{t^{(v-2)/2}e^{-t/2}}{2^{v/2}\Gamma(v/2)}\,\mathrm{d}t,$$

where $\Gamma(.)$ is the gamma function.

(3) *Normal pdf*: For each element of X, $y_{\text{norm\_pdf}}$ compute the pdf at X of the normal distribution with mean μ and variance σ as follows:

$$y_{\text{norm\_pdf}} = f(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}e^{-(x-\mu)^2/2\sigma^2}$$

(4) *Normal cdf*: For each element of X, compute the cdf at X of the normal distribution with mean μ and variance σ as follows:

$$p_{\text{norm\_cdf}} = F(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{x} e^{-(x-\mu)^2/2\sigma^2}\,\mathrm{d}t.$$

The standard normal distribution offers mean $\mu = 0$ and standard deviation $\sigma = 1$.

(5) *Binomial pdf*: For each element of X, $y_{\text{bio\_pdf}}$ compute the pdf at X of the binomial distribution with parameters n and p as follows:

$$y_{\text{bio\_pdf}} = f(x|n,p) = \binom{n}{x}p^x q^{(n-x)} I_{(0,1,\ldots,n)}(x).$$

(6) *Discrete uniform cdf*: The discrete uniform distribution is a simple distribution that puts equal weight on the integers from one to N. For each element of X, $p_{\text{du\_cdf}}$ compute the cdf at X of a univariate discrete distribution which assumes the values in x with probabilities P as follows:

$$p_{\text{du\_cdf}} = F(x|N) = \frac{\text{floor}(x)}{N} I_{(1,\ldots,N)}(x).$$

The values $\times$ do not need to be integers.

(7) *Exponential pdf*: For each element of X, $y_{\text{exp_pdf}}$compute the pdf of the exponential distribution with parameter μ as follows:

$$y_{\text{exp\_pdf}} = f(x|\mu) = \frac{1}{\mu}e^{x/\mu}.$$

(8) *F pdf*: For each element of X, y_{F_pdf} compute the pdf at X of the F distribution with v_1 and v_2 degrees of freedom as follows:

$$y_{F\_\text{pdf}} = f(x|v_1,v_2) = \frac{\Gamma[(v_1+v_2)/2]}{\Gamma(v_1/2)\Gamma(v_2/2)}\left(\frac{v_1}{v_2}\right)^{v_1/2} \times \frac{x^{v_1-2/2}}{[1+(v_1/v_2)x]^{(v_1+v_2)/2}}.$$

The parameters v_1 and v_2 must be positive integer. Non-central F probability density function ($y_{\text{ncf\_pdf}}$) measures the F pdf by considering a central parameter.

(9) *Gamma pdf*: For each element of X, $y_{\text{gama\_pdf}}$ return the pdf at X of the Gamma distribution with parameters a and b as follows:

$$y_{\text{gama\_pdf}} = f(x|a,b) = \frac{1}{b^a\Gamma(a)}x^{a-1}e^{x/b}.$$

The parameters a and b must be positive, and the values in X must lie between 0 and ∞.

(10) *Geometric cdf*: For each element of X, $p_{\text{gama\_cdf}}$ compute the cdf at X of the geometric distribution with parameter q as follows:

$$p_{\text{gama\_cdf}} = F(x|r) = \sum_{i=0}^{\text{floor}(x)} rq^i,$$

where $q = 1 - r$.

(11) *Hypergeometric cdf*: For each element of X, $p_{\text{hyp\_cdf}}$ compute the cdf at X of the hypergeometric distribution with parameters M, K, and N as follows:

$$p_{\text{hyp\_cdf}} = F(x|M,K,N) = \sum_{i=0}^{x}\frac{\binom{K}{i}\binom{M-K}{N-i}}{\binom{M}{N}}.$$

(12) *Lognormal pdf:* For each element of X, $y_{\text{log\_pdf}}$ compute the pdf at X of the lognormal distribution with parameters mean μ and variance σ as follows:

$$y_{\text{log\_pdf}} = f(x|\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}}\,e\,\frac{-(\ln(x)-\mu)^2}{2\sigma^2}.$$

(13) *Poisson pdf*: For each element of X, $y_{\text{poi\_pdf}}$ compute the pdf at X of the poisson distribution with parameter λ as follows:

$$y_{\text{poi\_pdf}} = f(x|\lambda) = \frac{\lambda^x}{x!}\,e^{-\lambda}I_{(0,1,\cdots)}(x),$$

where x can be any non-negative integer.

(14) *Rayleigh pdf*: For each element of X, $y_{\text{ray\_pdf}}$ computes the pdf at X of the Rayleigh distribution using the corresponding parameters in b as follows:

$$y_{\text{ray\_pdf}} = f(x|b) = b^{x/2}e^{(-x^2/2b^2)}.$$

The parameter b is in similar size with the vector or matrix X.

(15) *Student's t pdf*: For each element of X, $y_{\text{stt\_pdf}}$ compute the pdf at X of the T (Student) distribution with v degrees of freedom as follows:

$$y_{\text{stt\_pdf}} = f(x|v) = \frac{\Gamma((v+1)/2)}{\Gamma(v/2)}\,\frac{1}{\sqrt{v\pi}}\,\frac{1}{(1+(x^2/v))^{(v+1)/2}}.$$

The degrees of freedom v must be positive integer.

References

[1] S. Ali, Automated support vector learning algorithms, PhD Thesis, Monash University, 2005.

[2] S. Ali, K.A. Smith, On learning algorithm selection for classification, Int. J. Appl. Soft Comput. 6 (2) (2006) 119–138.

[3] K.P. Bennett, S. Wu, L. Auslender, On support vector decision trees for database marketing, IEEE International Joint Conference on Neural Networks (IJCNN '99) 2 (1999) 904–909.
[4] C. Blake, C.J. Merz, UCI Repository of Machine Learning Databases, University of California, Irvine, CA, 2002. http://www.ics.uci.edu/~mlearn/MLRepository.html
[5] B.E. Boser, I. Guyon, V.N. Vapnik, A Training algorithm for optimal margin classifiers, in: Proceedings of the Fifth Annual Workshop of Computational Learning Theory, vol. 5, ACM, Pittsburgh, 1992, pp. 144–152.
[6] C. Cortes, V. Vapnik, Support vector networks, Mach. Learning 20 (1995) 273–297.
[7] R.P.W. Duin, A note on comparing classifier, Pattern Recogn. Lett. 1 (1996) 529–536.
[8] T. Evgeniou, M. Pontil, T. Poggio, Regularization networks and support vector machines, Adv. Comput. Math. (1999).
[9] S.R. Gunn, Support Vector Machines for Classification and Regression, ISIS Technical Report, Image Speech and Intelligent Systems Group, University of Southampton, UK, 1998.
[10] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learning 46 (1/3) (2002) 389–422.
[11] K. Hyun-Chul, P. Shaoning, J. Hong-Mo, K. Daijin, B. Sung Yang, Pattern classification using support vector machine ensemble, in: Proceedings of IEEE 16th International Conference on Pattern Recognition, vol. 2, 2002, pp. 160–163.
[12] J.D. Jobson, Applied Multivariate Data Analysis: Categorical and Multivariate Methods, vol. II, Springer, New York, 1991.
[13] T.-S. Lim, Knowledge Discovery Central, Data Sets, http://www.KDCentral.com/, 2002.
[14] W. Mandenhall, T. Sincich, Statistics for Engineering and the Sciences, forth ed, Prentice-Hall, Englewood Cliffs, NJ, 1995.
[15] G. McLachlan, Discriminant Analysis and Statistical Pattern Recognition, Wiley, New York, 1992.
[16] A.M. Mood, F.A. Graybill, D.C. Boes, Introduction to the Theory of Statistics, third ed, McGraw-Hill, USA, 1974.
[17] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, B. Scholkopf, An introduction to kernel based learning methods, IEEE Trans. Neural Networks 12 (2) (2001) 181–201.
[18] P. Navarrete, J. Ruiz-del-Solar, Kernel-based face recognition by a reformulation of kernel machines, in: J. Benitez, F. Hoffmann (Eds.), Advances in Soft Computing—Engineering, Design and Manufacturing, Springer, Berlin, 2003, pp. 183–196.
[19] G. Paab, E. Leopold, M. Larson, J. Kindermann, S. Eickeler, SVM classification using sequences of phonemes and syllables, in: Proceedings of European Conference on Machine Learning (ECML), Helsinki, 2002.
[20] E. Parrado-Hernandez, I. Mora-Jimenez, J. Arenas-Garca, A.R. Figueiras-Vidal, A. Navia-Vazquez, Growing support vector classifiers with controlled complexity, Pattern Recogn. Lett. 36 (2003) 1479–1488.
[21] R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufman Publishers, San Mateo, CA, 1993.
[22] A. Shun-ichi, W. Si, Improving support vector machine classifiers by modifying kernel functions, Neural Networks 12 (1999) 783–789.
[23] K.A. Smith, F. Woo, V. Ciesielski, R. Ibrahim, Modelling the relationship between problem characteristics and data mining algorithm performance using neural networks, in: C. Dagli, et al. (Eds.), Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Complex Systems, vol. 11, ASME Press, New York, 2001, pp. 357–362.
[24] K.A. Smith, F. Woo, V. Ciesielski, R. Ibrahim, Matching data mining algorithm suitability to data characteristics using a self-organising map, in: A. Abraham, M. Koppen (Eds.), Hybrid Information Systems, Physica-Verlag, Heidelberg, 2002, pp. 169–180.
[25] Statistics Toolbox User's Guide, Version 3, The MathWorks, Inc., USA, 2001.
[26] A.C. Tamhane, D.D. Dunlop, Statistics and Data Analysis, Prentice-Hall, USA, 2000.
[27] K. Tsuda, G. Ratsch, S. Mika, K.-R. Muller, Learning to predict the leave-one-out error of kernel based classifiers, in: G. Dorffner, H. Bischof, K. Hornik (Eds.), Artificial Neural Networks—ICANN'01, vol. 2130, Springer Lecture Notes in Computer Science, New York, 2001, pp. 331–338.
[28] V. Vapnik, The Nature of the Statistical Learning Theory, Springer, New York, 1995.
[29] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[30] V.N. Vapnik, An overview of statistical learning theory, IEEE Trans. Neural Networks 10 (5) (1999).
[31] R. Vilalta, Y. Drissi, A perspective view and survey of meta-learning, Artif. Intell. Rev. 18 (2002) 77–95.
[32] V. Wan, S. Renals, Evaluation of kernel methods for speaker verification and identification, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02), vol. 1, 2002, pp. 6669–6672.
[33] J. Weston, C. Watkins, Multi-class support vector machines, in: M. Verleysen (Ed.), Presented at the Proceedings of the ESANN99, Brussels, Belgium, 1999.

Dr. Shawkat Ali is a lecturer in the School of Information Systems at Central Queensland University, Rockhampton, Australia. He holds a B.Sc. (Hons.) and M.Sc. in Applied Physics and Electronics, and M.Phil. in Computer Science and Technology from University of Rajshahi, Bangladesh and a Ph.D. in Information Technology from the Monash University, Australia. He was also an Assistant Professor at Islamic University, Bangladesh where he worked for 4 years prior to joining Monash University in 2001. Dr. Ali has published a quite good number of refereed journal and international conference papers in the areas of support vector machine, data mining and telecommunication.

Prof. Kate Smith-Miles is a Professor and Head of the School of Engineering and Information Technology at Deakin University in Australia. She obtained a B.Sc.(Hons) in Mathematics and a Ph.D. in Electrical Engineering, both from the University of Melbourne, Australia. She was also a Professor at Monash University, Australia, and co-Director of the Monash Data Mining Centre where she worked for ten years prior to joining Deakin University in 2006. Kate has published 2 books on neural networks and data mining in business, and over 150 refereed journal and international conference papers in the areas of neural networks, combinatorial optimisation, intelligent techniques and data mining. She has been awarded over AUD$1.5 million in competitive grants, including 7 Australian Research Council grants and industry awards. She is on the editorial board of several international journals, and has been a member of the organising committee for over 40 international data mining and neural network conferences. In addition to her academic activities, she also regularly acts as a consultant to industry in the areas of data mining and intelligent techniques.