Variant Methods of Reduced Set Selection for Reduced Support Vector Machines

Li-Jen Chien*, Chien-Chung Chang, Yuh-Jye Lee
Computer Science & Information Engineering
National Taiwan University of Science and Technology
Taipei 106, Taiwan
D8815002, D9115009, yuh-jye@mail.ntust.edu.tw

May 31, 2008

Abstract

In dealing with large datasets the reduced support vector machine (RSVM) was proposed for the practical objective to overcome the computational difficulties as well as to reduce the model complexity. In this paper, we propose two new approaches to generate representative reduced set for RSVM. First, we introduce Clustering Reduced Support Vector Machine (CRSVM) that builds the model of RSVM via RBF (Gaussian kernel) construction. Applying clustering algorithm to each class, we can generate cluster centroids of each class and use them to form the reduced set which is used in RSVM. We also estimate the approximate density for each cluster to get the parameter used in Gaussian kernel which will save a lot of tuning time. Secondly, we present Systematic Sampling RSVM (SSRSVM) that incrementally selects the informative data points to form the reduced set while the RSVM used random selection scheme. SSRSVM starts with an extremely small initial reduced set and adds a portion of misclassified points into the reduced set iteratively based on the current

*Corresponding Author.

classifier until the validation set correctness is large enough. We also show our methods, CRSVM and SSRSVM with smaller size of reduced set, have superior performance than the original random selection scheme.

Key words and phrases: kernel methods, kernel width estimation, Nyström approximation, reduced set, sampling methods, support vector machines.

1 Introduction

In recent years support vector machines (SVMs) with linear or nonlinear kernels [2, 6, 27] have become one of the most promising learning algorithms for classification as well as for regression [7, 18, 19, 25, 12], which are two fundamental tasks in data mining [29]. Via the use of kernel mapping, variants of SVM have successfully incorporated effective and flexible nonlinear models. There are some major difficulties that confront large data problems due to dealing with a fully dense nonlinear kernel matrix. To overcome computational difficulties some authors have proposed low-rank approximation to the full kernel matrix [24, 28]. As an alternative, Lee and Mangasarian have proposed the reduced support vector machine (RSVM) [14]. The key ideas of the RSVM are as follows. Prior to training, it randomly selects a portion of dataset as to generate a thin rectangular kernel matrix. Then it uses this much smaller rectangular kernel matrix to replace the full kernel matrix in the nonlinear SVM formulation. Computational time, as well as memory usage, is much less demanding for RSVM than that for a conventional SVM using the full kernel matrix. As a result, the RSVM also simplifies the characterization of the nonlinear separating surface. RSVM has comparable test errors, sometimes even slightly smaller. In other words, the RSVM has comparable, or sometimes slightly better, generalization ability. This phenomenon can be interpreted by the Minimum Description Length [20] as well as the Occam's razor [22].

Although the original random selection scheme has a good theoretical foundation [13], it may not be good representatives of the real data when the size of reduced set is too small [30]. Different strategies and many kind of basis selection methods have been discussed [10], in this paper, we propose two new approaches to generate the reduced set. The first method named Clustering Reduced Support Vector Machine (CRSVM) [5] that applies the k-means clustering algorithm to each class to generate cluster

centroids of each class and then use them to form the reduced set that is randomly selected in RSVM [14]. One of the most important ideas of SVM is kernel technique that uses a kernel function to represent the inner product of two data points in the feature space after a nonlinear mapping. We will use the Gaussian kernel through this paper. The value of the Gaussian kernel can be interpreted as a measure of similarity between data points. In this case, the reduced kernel matrix records the similarity between the reduced set and the entire training dataset. This observation inspires us to select the most representative points of the entire training dataset to form the reduced set. Using the cluster centroids would be intuitive heuristics. In order to catch the characteristic of each class we run the k-means clustering algorithm on each class separately. This idea originally comes from [16]. The Gaussian kernel function contains a tuning parameter σ, which determines the shape of the kernel function. Choosing this tuning parameter is called the model selection which is a very important issue in nonlinear support vector machine. In practice, the conventional SVM as well as RSVM determine this tuning parameter which is commonly used in kernel function via a tuning procedure [4]. While, in our approach the kernel width parameter is determined automatically for each point in the reduced set. This can be achieved by estimating the approximate density of each resulting cluster [21]. Once we have the reduced kernel matrix, we apply smooth support vector machine [14] to generate the final classifier. In the second approach, we use a systematic sampling mechanism to select a reduced set and name it as Systematic Sampling RSVM (SSRSVM) [3]. This algorithm is inspired by the key idea of SVM that the SVM classifier can be represented by support vectors and the misclassified points are a part of support vectors. The SSRSVM randomly selects an extremely small subset as an initial reduced set. Then, a portion of misclassified points are added into the reduced set iteratively based on the current classifier until the validation set correctness is large enough. We tested our methods, CRSVM and SSRSVM, on six public available datasets [1, 8] respectively. Under the compatible classification performance on the test set, CRSVM and SSRSVM can generate a smaller reduced set than the one via random selection scheme. Furthermore, experiments on real datasets present that CRSVM determines the kernel parameter automatically and individually for each point in the reduced set while the RSVM used a common kernel parameter which is determined by a tuning procedure. We also added the comparison of the

eigen-structures between the full kernel matrix, reduced kernel matrix via random selection, CRSVM and SSRSVM. The results have shown that the CRSVM and SSRSVM can provide good discriminant function estimations in supervised learning tasks. We also observe that CRSVM and SSRSVM are much faster than conventional SVM under the same level of the test set correctness. Although we focus on reduced set selection for SSVM, the same methods also can be applied to SSVR [12].

All notations used in the paper are listed as follows. All vectors will be column vectors unless otherwise specified or transposed to a row vector by a prime superscript $'$. The plus function x_+ is defined as $(x)_+ = \max\{0, x\}$. The scalar (inner) product of two vectors x and z in the n-dimensional real space R^n will be denoted by $x'z$ and the p-norm of x will be denoted by $\|x\|_p$. For a matrix $A \in R^{m\times n}$, A_i is the ith row of A which is *a row vector* in R^n. A column vector of ones of arbitrary dimension will be denoted by $\mathbf{1}$. For $A \in R^{m\times n}$ and $B \in R^{n\times l}$, the kernel $K(A, B)$ maps $R^{m\times n} \times R^{n\times l}$ into $R^{m\times l}$. In particular, $K(x', z)$ is a real number, $K(x', A')$ is a *row* vector in R^m, $K(A, x)$ is a *column* vector in R^m and $K(A, A')$ is an $m \times m$ matrix. The base of the natural logarithm will be denoted by e.

This paper is organized as follows. Section 2 provides the main ideas and formulation for RSVM. In section 3, we give a study on model selection of reduced kernels via centroid subset and the corresponding kernel width. Another method SSRSVM is described in section 4. The experimental results of our methods are presented in section 5 and section 6 concludes the paper.

2 Reduced Support Vector Machines

We now briefly describe the RSVM formulation, which is derived from the generalized support vector machine (GSVM) [17] and the smooth support vector machine (SSVM) [15]. We are given a training dataset $\{(x^i, y_i)\}_{i=1}^m$, where $x^i \in R^n$ is an input data point and $y_i \in \{-1, 1\}$ is class label, indicating one of two classes, A_- and A_+, to which the input point belongs. We represent these data points by an $m \times n$ matrix A, where the ith row of the matrix A, A_i, corresponds to the ith data point. We denote alternately A_i (a row vector) and x^i (a column vector) for the same ith data point. We use an $m \times m$ diagonal matrix D defined by $D_{ii} = y_i$ to specify the membership of each input point. The main goal of the classification problem is to find a classifier

that can predict the label of new unseen data points correctly. This can be achieved by constructing a linear or nonlinear separating surface, $f(x) = 0$, which is implicitly defined by a kernel function. We classify a test point x belonging to A_+ if $f(x) \geq 0$, otherwise x belonging to A_-. We will focus on the nonlinear case that is implicitly defined by a Gaussian kernel function. The RSVM solves the following unconstrained minimization problem

$$\min_{(\bar{v},\gamma)\in R^{\bar{m}+1}} \frac{\nu}{2}\|p(\mathbf{1} - D(K_\sigma(A,\bar{A}')\bar{v} - \mathbf{1}\gamma), \alpha)\|_2^2 + \frac{1}{2}(\bar{v}'\bar{v} + \gamma^2), \tag{1}$$

where the function $p(x, \alpha)$ is a very accurate smooth approximation to $(x)_+$ [15], which is applied to each component of the vector $\mathbf{1} - D(K_\sigma(A, \bar{A}')\bar{v} - \mathbf{1}\gamma)$ and is defined componentwise by

$$p(x, \alpha) = x + \frac{1}{\alpha}\log(1 + e^{-\alpha x}), \alpha > 0. \tag{2}$$

The function $p(x, \alpha)$ converges to $(x)_+$ as α goes to infinity. The reduced kernel matrix $K_\sigma(A, \bar{A}') \in R^{m\times\bar{m}}$ in (1) is defined by

$$K_\sigma(A, \bar{A}')_{ij} = e^{-\frac{||A_i - \bar{A}_j||_2^2}{2\sigma^2}}, \tag{3}$$

where $\bar{A}$ is the reduced set that is randomly selected from A in RSVM [15]. The positive tuning parameter ν here controls the tradeoff between the classification error and the suppression of $(\bar{v}, \gamma)$. Since RSVM has reduced the model complexity via using a much smaller rectangular kernel matrix we will suggest using a larger tuning parameter ν here. The solution of this minimization problem (1) for $\bar{v}$ and γ leads to the nonlinear separating surface

$$f(x) = \bar{v}'K_\sigma(\bar{A}, x) - \gamma = \sum_{i=1}^{\bar{m}} \bar{v}_i K_\sigma(\bar{A}_i, x) - \gamma = 0. \tag{4}$$

The minimization problem (1) can be solved via the Newton-Armijo method [15] directly and the existence and uniqueness of the optimal solution of this problem are also guaranteed. We note that this nonlinear separating surface (4) is a linear combination of a set of kernel functions $\left\{1, K_\sigma(\bar{A}_1, \cdot), K_\sigma(\bar{A}_2, \cdot), \cdots, K_\sigma(\bar{A}_{\bar{m}}, \cdot)\right\}$, where σ is the kernel parameter of each kernel function. In next section, we will apply

the k-means algorithm to each class to generate cluster centroids and then use these centroids to form the reduced set. Moreover we also give a formula to determine the kernel parameter σ for each point in the reduced set automatically.

In the following sections, we will introduce several different methods for generating more suitable reduced sets than the random selection scheme.

3 Clustering Reduced Support Vector Machine

We propose our new algorithm, Clustering RSVM (CRSVM), which combines the RSVM [14] and RBF networks algorithm together. We also describe how to estimate the kernel widths.

3.1 Parameters Estimation in RBFN

The most popular RBF networks can be described as

$$f(x) = w_0 + \sum_{h=1}^{\bar{m}} w_h \, e^{-\frac{||x - c^h||_2^2}{2\sigma_h^2}}, \tag{5}$$

where $c^h = (c_1^h, c_2^h, ..., c_n^h)$ is a vector in the n-dimensional vector space and $||x - c^h||_2$ is the distance between training (test) vectors x and c^h. We can use the same decision rule in previous section for binary classification. That is, we classify a test point x belonging to A_+ if $f(x) \geq 0$, otherwise x belonging to A_-. By RBFN approaches, we have to choose three parameters (c^h, σ_h ,w_h) in equation (5) based on the training dataset. For the first two parameters, many RBFN approaches were proposed that apply variant clustering algorithms such as k-means to training set to generate the cluster centroids as c^h. The parameter σ_h is estimated based upon the distribution of clusters. [21] estimates σ_h as

$$\sigma_h = \frac{\overline{R}(c^h) \cdot \delta \cdot \sqrt{\pi}}{\sqrt[n]{(r+1)\Gamma(\frac{n}{2}+1)}} \text{ , where } \delta \cdot \sqrt{\pi} = 1.6210 \tag{6}$$

and $\overline{R}(c^h)$ is defined as

$$\overline{R}(c^h) = \frac{n+1}{n}(\frac{1}{r}\sum_{q=1}^{r} ||\hat{x}_q - c^h||_2), \tag{7}$$

where $\hat{x}_1, \ldots, \hat{x}_r$ are the r nearest samples to the cluster centroid c^h. If the cluster size is smaller than r, we use the all examples in this cluster to compute $\bar{R}(c^h)$.

Since directly using parameter σ_h selected via RBF estimation did not perform very well, we tried to solve the problem based on the new kernel width estimation approach and adjusted the selected kernel width parameter σ_h to fall in the range of the new estimation.

3.2 Kernel Widths Estimation and Algorithm Description

We note that the width parameter σ is the key performance factor of SVMs model. Too large or too small σ value will lead to over-fitting or under-fitting respectively [11].

Based on the thought of density estimation, we apply r-nearest neighbor estimated algorithm to find the kernel width for centroids, in experiments we find that the class with fewer samples will have larger estimated kernel width parameters in average, and vice versa. Since the average distance between centroid and its r nearest neighbors seems to be dominated to the density of the cluster, it will be larger for sparse case and smaller for dense one. Using the proposed algorithm, for each centroid, we will get larger kernel width (σ) for clusters with the sparse data points, the shape of generated RBF would be smoother to cover a wider range of space and could be used to distinguish all the sparse points. For centroids in dense cluster, the shape of generated RBF should be sharper to just cover the dense points.

Since the original results are not well based on the default σ_h's estimation, we use a heuristic estimation in [9] to linearly interpolate in middle half of the search range of σ which is able to automatically scale the distance factor in Gaussian kernel. Let $A^*_{\bar{i}}$ and $A^*_{\bar{j}}$ be a pair of the *closest distinct points* in the training dataset and let $\rho = \|A^*_{\bar{i}} - A^*_{\bar{j}}\|_2^2$, i.e., $\rho = \min_{A_{\bar{i}} \neq A_{\bar{j}}} \|A_{\bar{i}} - A_{\bar{j}}\|_2^2$. We confine the kernel function value of this pair of points to the range $[0.150, 0.999]$. That is

$$0.150 \leq e^{-\frac{|A^*_{\bar{i}} - A^*_{\bar{j}}\|_2^2}{2\sigma^2}} = e^{-\frac{\rho}{2\sigma^2}} \leq 0.999. \tag{8}$$

In practice, finding the centroids and *closest distinct points* in a massive training dataset is very time consuming. We suggest the follow scheme for the upper and

lower bound estimates based on a random subset. First, randomly sample a small subset from the entire dataset, then calculate the upper and lower bounds using this random subset, and finally adjust the bounds by a multiplicative factor $(m/\bar{m})^{2/(4+d)}$, where $\bar{m}$ is the subset size and d is the dimension of x.[1]

Based on the interpolation, we can calculate σ_{h_n} from σ_h mentioned in (6), the tuning parameter left in RSVM is only ν. We proposed a variant RSVM method that uses clustering centroids as reduced set. The Clustering Reduced Support Vector Machine (CRSVM) algorithm is described below.

Algorithm 3.1 Clustering Reduced Support Vector Machine

Let k be the number of cluster centroids for each class and r be a positive integer.

Step 1. For each class, runs k-means algorithm to find the cluster centroids c^h. Use the clustering results to form the reduced set $\bar{A} = [c^1 c^2 ... c^{2k}]'$.

Step 2. For each centroid c^h, computes the corresponding kernel parameter σ_{h_n}.

Step 3. Let A_i denotes the ith training point, use the resulting parameters from *Step* 1 and *Step* 2 to construct the rectangular kernel matrix $K_\sigma(A, \bar{A}')_{ih} = e^{-\frac{||A_i - c^h||_2^2}{2{\sigma_{h_n}}^2}}$, where $K_\sigma \in R^{m \times 2k}$, for $i = 1, 2, ..., m$ and $j = 1, 2, ..., 2k$.

Step 4. Apply the Newton-Armijo Algorithm [14] to solve the problem (1), where $K_\sigma(A, \bar{A}')$ is the reduced kernel matrix obtained in *Step* 3.

Step 5. The separating surface is given as formula (4), where $(\bar{v}^*, \gamma^*) \in R^{\bar{m}+1}$ is the unique solution of problem (1) that got from *Step* 4.

Step 6. A new unseen data point $x \in R^n$ is classified as class +1 if $\bar{v^*}' K_\sigma(\bar{A}, x) - \gamma^* \geq 0$, otherwise x is classified as class -1.

[1]For assessing the search range of σ using a reduced set, it should be adjusted accordingly to account for the effect caused by using only a fraction $\bar{m}/m$ of data. It is well known in the nonparametric literature that an ideal window width σ is of order $\sigma = O(m^{-1/(4+d)})$ (cf. Stone [26] and Silverman [23].) Thus, if only a fraction $\bar{m}/m$ of data is used, a multiplicative factor $(m/\bar{m})^{2/(4+d)}$ adjustment should be adopted.

The conventional SVMs as well as RSVM determine parameter used in kernel function via a tuning procedure. While, in our approach the kernel parameter is determined automatically and individually for each point in the reduced set. This can be achieved by estimating the approximate density of each resulting cluster [21]. For large datasets, learning will take a long time. We can randomly choose subset from training set and stop the k-means algorithm at 5 iterations to save the learning time. The numerical results are showed in subsection 5.2.

4 Systematic Reduced Set Selection

We now introduce another new algorithm to generate the reduced set which is consisting of the *informative* data points. This algorithm is inspired by the key idea of SVM, the SVM classifier can be represented by support vectors and the misclassified points are a part of support vectors. Instead of random sampling the reduced set in RSVM, we start with an extremely small initial reduced set and add a portion of misclassified points into the reduced set iteratively based on the current classifier. We note that there are two types of misclassified points and we select them respectively and show this idea in Fig. 1. The new reduced kernel matrix can be updated from the previous iteration. We only need to augment the columns which are generated by the new points in the reduced set. We stop this procedure until the validation set correctness is large enough.

Algorithm 4.1 Systematic Sampling RSVM Algorithm

Step 1. Randomly select an extremely small portion data points, such as $\bar{m} = 5$, from the entire training data matrix $A \in R^{m \times n}$ as an initial reduced set which is represented by $\bar{A}_0 \in R^{\bar{m} \times n}$.

Step 2. Generate the reduced kernel matrix $K(A, \bar{A}_0')$ and perform RSVM algorithm [14] to generate a tentative separating surface represented by $f(x) = 0$.

Step 3. Use the separating surface to classify the point which is in the training set but not in the current reduced set. Let $\bar{I}_+$ be the index set of misclassified points of positive example. That is, $\bar{I}_+ = \{i | f(A_i) \leq 0, A_i \in A_+\}$. Similarly, $\bar{I}_- = \{i | f(A_i) > 0, A_i \in A_-\}$.

Step 4. Sort the set $\bar{I}_+$ by the absolute value of $f(A_{\bar{I}_+})$ and the set $\bar{I}_-$ by $f(A_{\bar{I}_-})$ respectively. We named the resulting sorted sets $\bar{S}_+$ and $\bar{S}_-$.

Step 5. Partition $\bar{S}_+$ and $\bar{S}_-$ into several subsets respectively such that each subset has nearly equal number of elements just like Fig. 1. That is, let $\phi \neq \bar{sp}_i \subset \bar{S}_+$, $\forall i, 1 \leq i \leq k$ where k is the number of subsets. $\bar{S}_+ = \bar{sp}_1 \cup \bar{sp}_2 \cup \ldots \cup \bar{sp}_k$ and $\bar{sp}_i \cap \bar{sp}_j = \phi, \forall i \neq j, 1 \leq i, j \leq k$. Similarly, $\bar{S}_- = \bar{sn}_1 \cup \bar{sn}_2 \cup \ldots \cup \bar{sn}_k$ and $\bar{sn}_i \cap \bar{sn}_j = \phi, \forall i \neq j, 1 \leq i, j \leq k$. Then, choose one point from each subset and add these points into $\bar{A}_0$ to generate a new reduced set in place of $\bar{A}_0$.

Step 6. Repeat *Step* $2 \sim 5$ until the validation set correctness has arrived at the threshold which is user pre-specified.

Step 7. Output the final classifier, $f(x) = 0$.

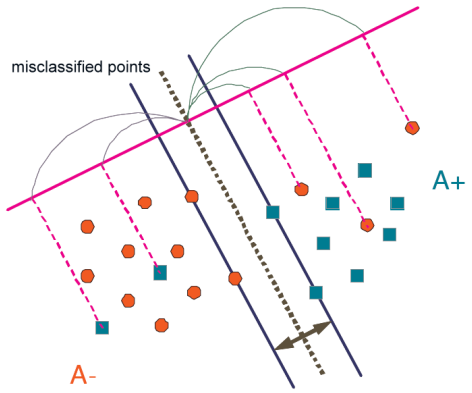

Figure 1: Illustrate the idea of Systematic Sampling RSVM Algorithm.

As mentioned above, the misclassified points can be treated as parts of support vectors and added into the reduced set. Thus for taking the information of the misclassified points uniformly, we use the systematic sampling approach to select the same number of misclassified points in different distance from them to separating surface. This incremental model selection scheme can prevent to add too many misclassified points in the same distance level and catch represented misclassified points in the reduced set of the new model as soon as possible. We showed the numerical results to demonstrate the efficiency of this algorithm in subsection 5.2.

5 Numerical Results

5.1 Spectral analysis

In this subsection, we attempt to explain why reduced kernel SVMs can perform successful as well as full kernel SVM from a point of view of spectral analysis. In order to avoid dealing with the huge and dense full kernel matrix in SVM, a low-rank approximation to the full kernel matrix which is known as the Nyström approximation has been proposed in many sophisticated ways [24, 28]. That is,

$$K(A, A') \approx K(A, \tilde{A}')K(\tilde{A}, \tilde{A}')^{-1}K(\tilde{A}, A') = \tilde{K}. \tag{9}$$

We denote the Nyström approximation of $K(A, A')$ by $\tilde{K}$ in the rest of this paper. Applying this approximation, for a vector $v \in R^m$,

$$K(A, A')v \approx K(A, \tilde{A}')K(\tilde{A}, \tilde{A}')^{-1}K(\tilde{A}, A')v = K(A, \tilde{A}')\tilde{v}, \tag{10}$$

where $\tilde{v} = K(\tilde{A}, \tilde{A})^{-1}K(\tilde{A}, A')v$. In the variant RSVM scheme, $\tilde{v}$ is directly determined by fitting the entire dataset. We generate the Nyström approximation from reduced sets sampled with random and k-means clustering selection scheme, and there are little differences in eigenvalues from the full kernel matrix. In order to have a better understanding of the differences of their spectral behaviors, we present six plots for four datasets. In Figs. 2-5, the horizontal axis N is the number of the eigenvalues listed. The left part of figures are based on low-rank approximation from 5% reduced kernels and the right part are based on 3% reduced kernels. We explain the left part of the figures and the right part is in a similar way. In Figs. 2(a) and (c)-5(a) and (c), we plot the eigenvalues of the full and the approximation kernels. We split them into two plots (a) and (c) due to their different scale. In Figs. 2(e)-5(e), the differences between the eigenvalues of the full and the approximation kernels which are generating from random scheme and clustering scheme are plotted against N.

From Fig. 2-5, we can observe that the quality of the approximation will depend on the rate of decay of the eigenvalues of the full kernel matrix (green circle) based on the same kernel width. The numerical simulations indicate that the reduced kernel generated by the clustering scheme (red star) retains the fewer differences with the full kernel than the traditional random selection scheme (blue diamond), even with fewer elements in reduced set. The differences of eigenvalues between full kernel and

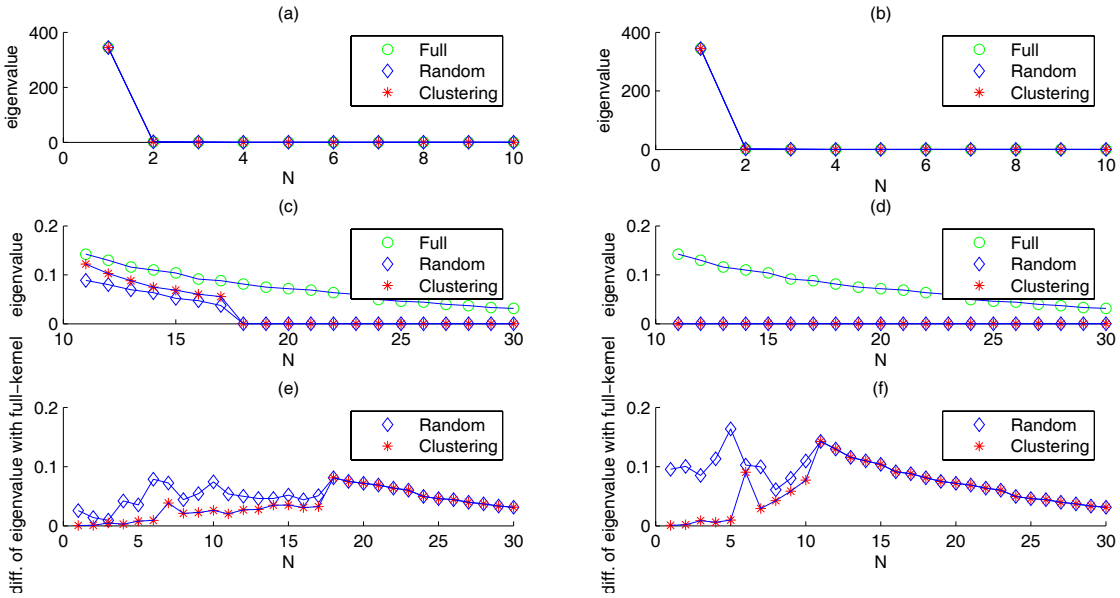

Figure 2: The spectral analysis of Ionosphere dataset

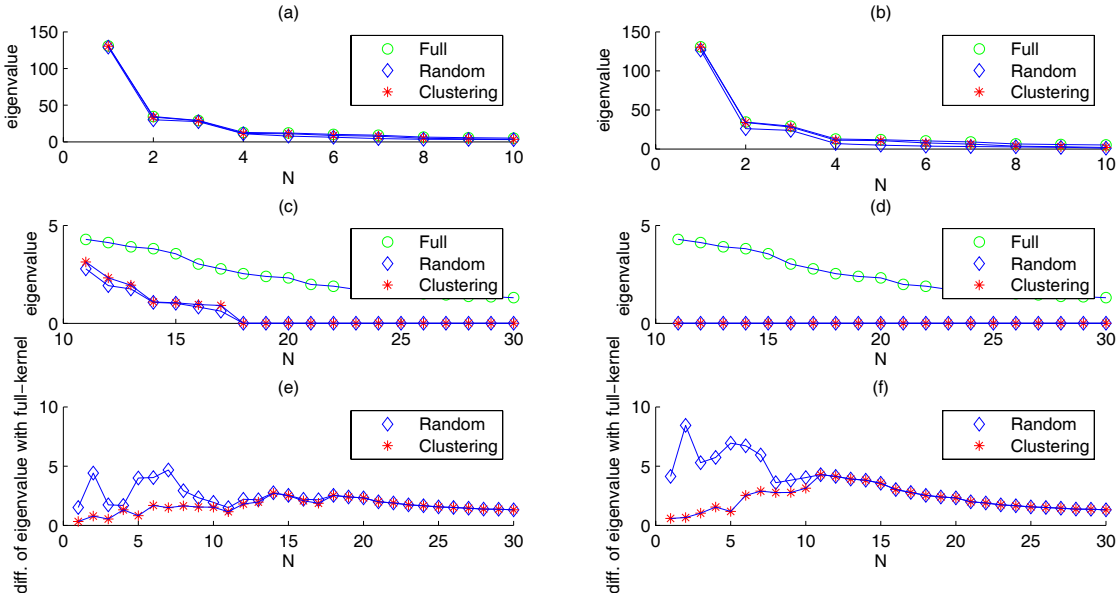

Figure 3: The spectral analysis of BUPA dataset

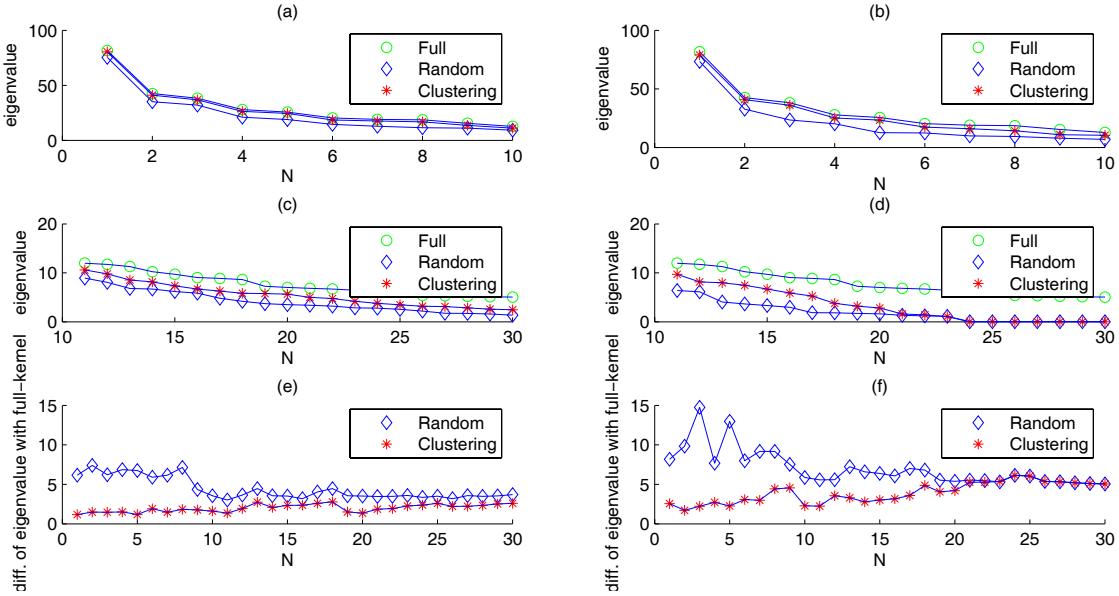

Figure 4: The spectral analysis of Pima dataset

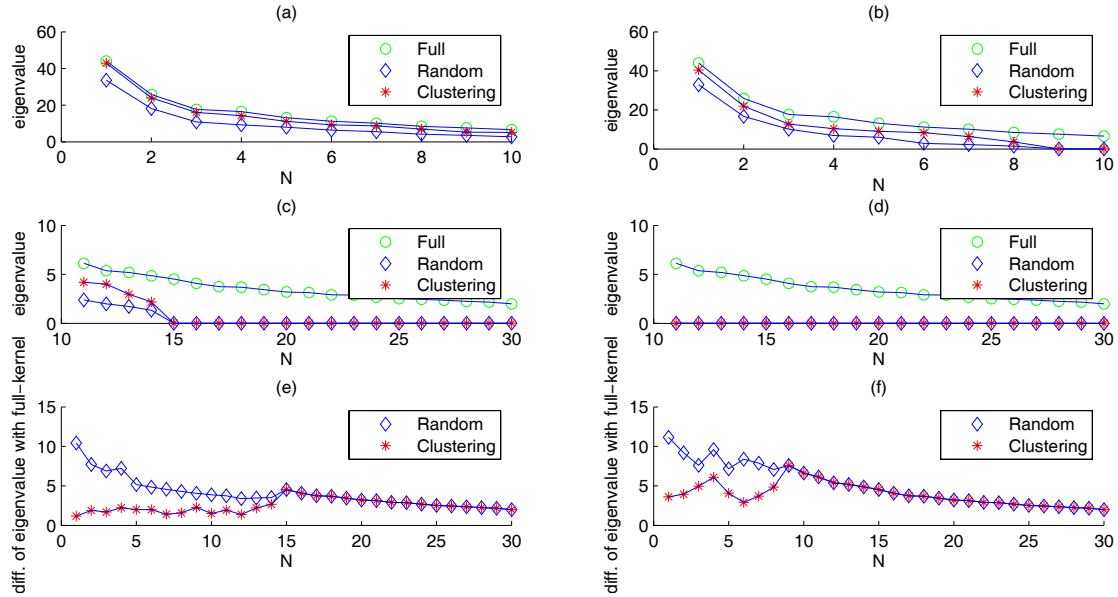

Figure 5: The spectral analysis of Cleveland dataset

approximation ones are all very small. These observations might give an explanation why the CRSVM can provide good discriminant function estimations in supervised learning tasks. This result indicate that the low rank approximation is good with small reduced set.

5.2 Comparison of Variant RSVM schemes

All our experiments were performed on a personal computer, which utilizes a 1.47 GHz AMD Athlon(tm) XP 1700 PLUS processor and 256 megabytes of RAM. This computer runs on Windows XP operating system, with MATLAB 6 installed. We implemented the CRSVM and SSRSVM algorithm using standard native MATLAB codes. We used the Gaussian kernel in all our experiments. We test CRSVM, SSRSVM and other SVMs on six public available datasets which five are from UC Irvine repository [1] and one is from MIT CBCL [8]. In order to give a more objective comparison, we run tenfold cross-validation on each dataset. All parameters in our experiments were chosen for optimal performance on a tuning set and kernel parameter selected in CRSVM are also adjusted in the estimated range of the tuning set, a surrogate for a test set. The computer ran out of memory while generating the full nonlinear kernel for the Mushroom and Face datasets. $\bar{m}$ denotes the average size of reduced set by running the SSRSVM algorithm. N/A denotes "not available" results because the kernel $K(A, A')$ was too large to store. We also report the number of support vectors in LibSVM's result for each dataset to be compared with the size of reduced set.

In all numerical tests in Table 1, the size of reduced set is smaller than the number of support vectors resulted from LibSVM. This indicates that RSVMs use fewer kernel bases to generate the discriminant function. RSVMs tend to have a simpler model and it needs a smaller number of function evaluations when predicting a new unlabeled data point. This is an advantage in the testing phase of learning tasks. Moreover, the numerical results demonstrated that SSRSVM can keep as good test set correctness as SSVM and RSVM and usually has less size of reduced set and time cost than RSVM. CRSVM save the most tuning time with kernel parameter estimation with fewer elements in reduced set and also keep the test correctness with RSVM. Table 1 summarizes the numerical results and comparisons of our experiments. It shows a

comparison on the testing correctness and time cost of CRSVM, SSRSVM, RSVM and SSVM algorithms.

Tenfold Test Set Correctness % Tenfold Computational Time, *Seconds*									
Dataset Size $m \times n$	Methods								
	CRSVM		SSRSVM		RSVM		SSVM	LibSVM	
	Correctness Time *sec.*	$\bar{m}$	Correctness Time *sec.*	$\bar{m}$	Correctness Time *sec.*	$\bar{m}$	Correctness Time *sec.*	Correctness Time *sec.*	#SV
Ionosphere 351×34	95.88(96.76) 0.3870	20	**97.43** 0.5620	20	96.87 0.6410	35	96.61 14.2190	95.16 0.1720	67.9
Cleveland Heart 297×13	84.36(86.20) 0.3621	20	86.20 0.5620	20.6	85.94 0.3750	30	**86.61** 7.2500	85.86 3.5460	140.6
BUPA Liver 345×6	72.75(73.23) 0.3081	18	74.80 0.4680	17.8	**74.87** 0.5000	35	74.47 10.1560	73.64 0.4620	216.2
Pima Indians 768×8	**78.11**(78.42) 0.6127	17	78.00 0.9690	17.4	77.86 1.5160	50	77.34 68.1560	75.52 26.8440	410.1
Mushroom 8124×22	88.41(89.06) 54.5073	79	89.23 74.6870	79	**89.39** 171.2500	215	N/A N/A	89.19 171.4840	1803.5
Face 6977×361	97.2(98.2) 58.4911	42	**98.51** 73.8120	42.2	98.39 115.2660	70	N/A N/A	98.15 318.9400	404.3

Table 1: Tenfold cross-validation correctness results on six public datasets. The best result is in boldface for each dataset. N/A Indicates the result is not available because of computational difficulties. We list the best correctness tuned by different search range from kernel with estimation for CRSVM in parentheses.

6 Conclusion

In this paper we propose two new approaches to generate the reduced set. One is CRSVM which builds the model of RSVM via RBF (Gaussian) kernel construction. Applying clustering algorithm to each class, we can generate cluster centroids of each class and use them to form the reduced set in RSVM. We also estimate the approximate density for each cluster to get the kernel parameter that is used in Gaussian kernel. By this way, we can save a lot of tuning time. Another is SSRSVM that selects the informative data points to form the reduced set iteratively while the RSVM used random selection scheme. This algorithm is inspired by the key idea of SVM, the SVM classifier can be represented by support vectors and the misclassified points are a part of support vectors. SSRSVM starts with an extremely small initial reduced

set and adds a portion of misclassified points into the reduced set iteratively based on the current classifier until the validation set correctness is large enough. In our experiments, we tested our methods, CRSVM and SSRSVM, on six public available datasets [1, 8] respectively. Under the compatible classification performance on the test set, CRSVM and SSRSVM can generate a smaller reduced set than the one via random selection scheme. CRSVM can determine the different kernel width parameters automatically for each point in the reduced set while the RSVM used a common kernel parameter which is determined by a tuning procedure. Performing very well averagely, SSRSVM is usually faster than RSVM and much faster than conventional SVM. For providing a better understanding of the reduced kernel technique, we also study the k-means clustering scheme and random selection scheme from a robust design point of view and measure the discrepancy between the full kernel by generating their Nyström approximation. It showed that k-means clustering scheme has better approximation results even with fewer elements in reduced sets. All results show that our CRSVM and SSRSVM can provide good discriminant function estimations via smaller reduced sets than the traditional random selection scheme in supervised learning tasks. We can also benefit from saving a lot of cost in training and testing stages.

References

[1] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[2] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[3] C.-C. Chang. and Y.-J. Lee. Generating the reduced set by systematic sampling. In *Proc. 5th Intelligent Data Engineering and Automated Learning*, pages 720–725, Exeter, UK, August 2004. LNCS 3177, Springer-Verlag.

[4] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1):131–159, 2002.

[5] L.-J. Chien. and Y.-J. Lee. Clustering model selection for reduced support vector machines. In *Proc. 5th Intelligent Data Engineering and Automated Learning*, pages 714–719, Exeter, UK, August 2004. LNCS 3177, Springer-Verlag.

[6] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, 2000.

[7] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems -9-*, pages 155–161, Cambridge, MA, 1997. MIT Press.

[8] MIT Center for Biological and Computation Learning. Cbcl face database (1), 2000. http://www.ai.mit.edu/projects/cbcl.

[9] C.-M. Huang, Y.-J. Lee, D. K. J. Lin, and S.-Y. Huang. Model selection for support vector machines via uniform design. *A special issue on Machine Learning and Robust Data Mining of Computational Statistics and Data Analysis*, 52:335–346, 2007.

[10] S. S. Keerthi, O. Chapelle, and D. DeCoste. Building support vector machines with reduced classifier complexity. *JMLR*, 7:1493–1515, 2006.

[11] S. S. Keerthi and C.-J. Lin. Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural Computation*, 15(7):1667–1689, 2003.

[12] Y.-J. Lee, W.-F. Hsieh, and C.-M. Huang. ϵ-SSVR: A smooth vector machine for ϵ-insensitive regression. *IEEE Transactions on Knowledge and Data Engineering*, 17:678–685, 2005.

[13] Y.-J. Lee and S.-Y. Huang. Reduced support vector machines: A statistical theory. *IEEE Transactions on Neural Networks*, 18(1):1–13, 2007.

[14] Y.-J. Lee and O. L. Mangasarian. RSVM: Reduced support vector machines. Technical Report 00-07, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, July 2000. Proceedings of the First SIAM International Conference on Data Mining, Chicago, April 5-7, 2001, CD-ROM Proceedings. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/00-07.ps.

[15] Y.-J. Lee and O. L. Mangasarian. SSVM: A smooth support vector machine. *Computational Optimization and Applications*, 20:5–22, 2001. Data Mining Institute, University of Wisconsin, Technical Report 99-03. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/99-03.ps.

[16] A. Lyhyaoui, M. Martinez, I. Mora, M. Vazquez, J.-L. Sancho, and A. R. Figueiras-Vidal. Sample selection via clustering to construct support vector-like classifier. *IEEE Transactions on Neural Networks*, 10:1474–1481, 1999.

[17] O. L. Mangasarian. Generalized support vector machines. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 135–146, Cambridge, MA, 2000. MIT Press. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-14.ps.

[18] O. L. Mangasarian and D. R. Musicant. Large scale kernel regression via linear programming. Technical Report 99-02, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, August 1999. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/99-02.ps.

[19] O. L. Mangasarian and D. R. Musicant. Robust linear and support vector regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):950–955, 2000. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/99-09.ps.

[20] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Boston, 1997.

[21] Y.-J. Oyang, S.-C. Hwang, Y.-Y. Ou, C.-Y. Chen, and Z.-W. Chen. An novel learning algorithm for data classification with radial basis function networks. In *Proceeding of 9th International Conference on Neural Information Processing*, pages 18–22, Singapore, Nov. 2001.

[22] C. E. Rasmussen and Z. Ghahramani. Occam's razor. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 294–300, Cambridge, MA, 2001. MIT Press.

[23] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman, Hall, London, 1986.

[24] A. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In *Proc. 17th International Conf. on Machine Learning*, pages 911–918. Morgan Kaufmann, San Francisco, CA, 2000.

[25] A. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14:199–222, 2004.

[26] C. J. Stone. An asymptotically optimal window selection rule for kernel density estimates. *Annals of Statistics*, 12:1285–1297, 1984.

[27] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[28] C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 682–688, Cambridge, MA, 2001. MIT Press.

[29] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco, CA, 1999.

[30] M. Wu, B. Schölkopf, and G. Bakir. A direct method for building sparse kernel learning algorithms. *Journal of Machine Learning Research*, 7:603–624, 2006.