# Incremental Reduced Support Vector Machines

Yuh-Jye Lee, Hung-Yi Lo
Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan 106
Email: {yuh-jye, M9115006}@mail.ntust.edu.tw

Su-Yun Huang
Institute of Statistical Science
Academia Sinica
Taipei, Taiwan 115
Email: syhuang@stat.sinica.edu.tw

**Abstract**

The reduced support vector machine (RSVM) has been proposed to avoid the computational difficulties in generating a nonlinear support vector machine classifier for a massive dataset. RSVM selects a small random subset from the entire dataset with a user pre-specified size $\bar{m}$ to generate a reduced kernel (rectangular) matrix. This reduced kernel will replace the fully dense square kernel matrix used in the nonlinear support vector machine formulation to cut the problem size and computational time and will not scarify the prediction accuracy. In this paper, we propose a new algorithm, Incremental Reduced Support Vector Machine (IRSVM). In contrast to purely random selection scheme used in RSVM, IRSVM begins with an extremely small reduced set and incrementally expands the reduced set according to an information criterion. This information-criterion based incremental selection can be achieved by solving a series of small least squares problems. In our approach, the size of reduced set will be determined automatically and dynamically but not pre-specified. The experimental tests on four publicly available datasets from the University of California (UC) Irvine repository show that IRSVM used a smaller reduced set than RSVM without scarifying classification accuracy.

**Keywords:** Support vector machine, reduced support vector machine, reduced set, kernel function, least squares problem.

## 1. Introduction

Recently, support vector machines (SVMs) with linear or nonlinear kernels [1], [2], [16] have become the most promising learning algorithm for classification as well as regression [3], [12] which are the fundamental tasks in Data Mining [17]. The SVM classifiers can be generated via solving a minimization problem. However, SVM suffers from the difficulty of long computational time and large memory usage in using nonlinear kernels on large datasets which come from many real applications. The reduced support vector machine (RSVM) [7] has been proposed to avoid these difficulties in generating a nonlinear support vector machine classifier for a massive dataset. The basic idea of RSVM is using a small rectangular kernel matrix to replace the fully dense square kernel matrix used in the nonlinear support vector machine formulation without scarifying the accuracy. Computational time, as well as memory usage, is much smaller for RSVM than that for a conventional SVM using the entire dataset. As a result, RSVM also simplifies the characterization of the nonlinear separating surface. According to Occam's razor [13], as well as Minimum Description Length (MDL) [13], RSVM might have better generalization ability than a conventional SVM. This reduced kernel technique has been successfully applied to other kernel-based learning algorithm, such as proximal support vector machine (PSVM) [4] and $\epsilon$-smooth support vector regression ($\epsilon$-SSVR) [8]. In [7], the reduced set is selected randomly from the entire dataset with a user pre-specified reduced set size $\bar{m}$. It is typically much smaller than the size of entire dataset. It is natural to raise two questions as follows:

1) Is there a way to choose the reduced

set other than random selection so that RSVM will have a better performance?
2) Is there a mechanism that determines the size of reduced set automatically or dynamically?

In this paper, we propose our Incremental Reduced Support Vector Machine (IRSVM) that automatically and incrementally selects representative data points to form the reduced set. The experimental tests on four publicly available datasets from the University of California (UC) Irvine repository [14] show that IRSVM used a smaller reduced set than RSVM without scarifying classification accuracy.

Now we briefly outline the contents of the paper. Section 2 provides the main idea and formula for RSVM. In section 3, we describe our sequential and batch versions of incremental reduced support vector machine. The numerical results are shown in Section 4. Section 5 concludes the paper.

A word about our notation and background material is given below. All vectors will be column vectors unless transposed to a row vector by a prime superscript $'$. For a vector $x$ in the $n$-dimensional real space $R^n$, the plus function $x_+$ is defined as $(x_+)_i = \max\{0, x_i\}$, while the step function $x_*$ is defined as $(x_*)_i = 1$ if $x_i > 0$ else $(x_*)_i = 0$, $i = 1, \ldots, n$. The scalar (inner) product of two vectors $x$ and $y$ in the $n$-dimensional real space $R^n$ will be denoted by $x'y$ 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 $i$th row of $A$ which is a row vector in $R^n$. A column vector of ones of arbitrary dimension will be denoted by $e$. 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, if $x$ and $y$ are column vectors in $R^n$ then, $K(x', y)$ is a real number, $K(x', A')$ is a *row* vector in $R^m$ and $K(A, A')$ is an $m \times m$ matrix. The difference of two sets $A$ and $B$ is defined as $A \setminus B = \{x | x \in A \text{ and } x \notin B\}$.

## 2. Reduced Support Vector Machines

We consider the problem of classifying points into two classes, $A_+$ and $A_-$. We are given a dataset consisting of $m$ points in the $n$-dimensional real space $R^n$. Each point in the dataset comes with a class label, $+1$ or $-1$, indicating one of two classes, $A_+$ and $A_-$, to which the point belongs. We represent these data points by an $m\times n$ matrix $A$, where the $i$th row of the matrix $A$, $A_i$, corresponds to the $i$th data point. We use an $m \times m$ diagonal matrix $D$ with ones or minus ones along its diagonal to specify the membership of each point. In other words, $D_{ii} = \pm 1$ depending on whether the label of $i$th data point is $+1$ or $-1$. 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 which is implicitly defined by a kernel function. We will focus on nonlinear case in this paper. The nonlinear kernel matrix $K(A, A') \in R^{m\times m}$ (where $m$ is the size of the training set) on large datasets used in conventional support vector machine [1], [2], [16] will lead to some computational difficulties [7]. To avoid these difficulties, the reduced support vector machine (RSVM) [7] uses a very small random subset of size $\bar{m}$ of the original $m$ data points, where $\bar{m} << m$. We denote this random subset by $\bar{A}$, which is used to generate a much smaller rectangular matrix $K(A, \bar{A}') \in R^{m\times\bar{m}}$ and to replace the huge and fully dense square kernel matrix $K(A, A')$ used in conventional SVM to cut problem size, computational time and memory usage as well as to simplify the characterization of nonlinear separating surface. We now briefly describe the reduced support vector machine formulation, which is derived from the generalized support vector machine (GSVM) [11] and smooth support vector machine [9]. The RSVM solves the following unconstrained minimization problem for an arbitrary rectangular kernel $K(A, \bar{A}')$:

$$\min_{(\bar{u},\gamma)\in R^{\bar{m}+1}} \frac{\nu}{2}\|p(e - D(K(A, \bar{A}')\bar{D}\bar{u} - e\gamma), \alpha)\|_2^2 + \frac{1}{2}(\bar{u}'\bar{u} + \gamma^2), \quad (1)$$

where the function $p(x, \alpha)$ is a very accurate smooth approximation to $(x)_+$ [9], which is applied to each component of the vector $e - D(K(A, \bar{A}')\bar{D}\bar{u} - e\gamma)$ and is defined componentwise by

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

The function $p(x, \alpha)$ converges to $(x)_+$ as $\alpha$ goes to infinity. The positive tuning parameter $\nu$ here controls the tradeoff between the classification error and the suppression of $(\bar{u}, \gamma)$.The diagonal matrix $\bar{D} \in R^{\bar{m}\times\bar{m}}$ with ones or minus ones along its diagonal to specify the membership of each point in the reduced set. A solution of this minimization program for $\bar{u}$ and $\gamma$ leads to the nonlinear separating surface

$$K(x', \bar{A}')\bar{D}\bar{u} = \gamma. \tag{3}$$

Problem (1) retains the strong convexity and differentiability properties in the $R^{\bar{m}+1}$ space of $(\bar{u}, \gamma)$ for any arbitrary *rectangular* kernel. Hence we can apply the Newton-Armijo Algorithm [9] directly to solve (1) and the existence and uniqueness of the optimal solution of the minimization problem (1) are also guaranteed. For convenience, we use $\Phi_{\alpha,\nu}(\bar{u}, \gamma)$ to represent the objective function (1) throughout the rest of the paper. We note that the nonlinear separating surface (3) is a linear combination of a set of kernel functions $\left\{1, K(\cdot, \bar{A}'_1), K(\cdot, \bar{A}'_2), \cdots, K(\cdot, \bar{A}'_{\bar{m}})\right\}$. That is, the separating surface is of the form

$$\sum_{i=1}^{\bar{m}} K(x', \bar{A}'_i)\bar{D}_{ii}\bar{u}_i = \gamma. \tag{4}$$

In a nutshell, the RSVM can be split into two parts. First, it selects a small random subset from the entire dataset to form a dictionary function set, $\left\{1, K(\cdot, \bar{A}'_1), K(\cdot, \bar{A}'_2), \cdots, K(\cdot, \bar{A}'_{\bar{m}})\right\}$. The size of this small random subset is prespecified by users. Secondly, it determines the best coefficients of the kernel functions in the dictionary function set by solving the unconstrained minimization problem (1). The classifier is a linear combination of these kernel functions with these coefficients. In next section we modify the first part of the RSVM algorithm and introduce an incremental approach that begins with an extremely small reduced set and then sequentially expands the reduced set according to an information criterion. The dictionary function set is generated by this resulting reduced set. Besides, the size of the dictionary function set is dynamically determined by the algorithm and typically is smaller than the purely random ones under the same performance level.

## 3. Incremental Reduced Support Vector Machines

In this section, we propose our Incremental Reduced Support Vector Machine (IRSVM) algorithm that automatically and incrementally selects informative data points from the entire training set to generate the rectangular kernel matrix used in RSVM.

The nonlinear classifier generated by RSVM, as shown in (4), is a linear combination of a dictionary function set [6] that consists of kernel functions $\left\{1, K(\cdot, \bar{A}'_1), K(\cdot, \bar{A}'_2), \cdots, K(\cdot, \bar{A}'_{\bar{m}})\right\}$ induced by the reduced set $\bar{A}$. Intuitively, if the kernel functions in the dictionary function set are very "*similar*", the hypothesis space spanned by this dictionary function set will be very limited.

Based on this intuition, we propose a process that sequentially adding a kernel function into the dictionary function set, only when the function is "*unsimilar*" to the current set and carrying sufficient extra information over the current set. We start with a very small reduced set $\bar{A}$, typically a size of 2, then we add a new data point $A_i$ into the reduced set only when the extra information carried in the vector $K(A, A'_i)$ with respect to the column space of $K(A, \bar{A}')$ is greater than a certain positive threshold. This can be achieved by solving a least squares problem. We let $\bar{K} = K(A, \bar{A}') \in R^{m\times\bar{m}}$ for the reason of convenience. The least squares problem we need to solve is

$$\min_{\beta\in R^{\bar{m}}} \left\|\bar{K}\beta - K(A, A'_i)\right\|_2^2, \tag{5}$$

where $\beta \in R^{\bar{m}}$ is a free vector variable and $\bar{K}\beta \in R^m$ is a linear combination of the functions $K(A, \bar{A}'_i), i = 1, \ldots, \bar{m}$ that represents the column space of $K(A, \bar{A}')$. According to the first order optimality condition [10], finding out the optimal solution $\beta^*$ of above problem (5) is equivalent to solving a system of normal equations:

$$\bar{K}'\bar{K}\beta = \bar{K}'K(A, A'_i). \tag{6}$$

If the columns of the rectangular kernel matrix generated by the initial reduced set are linear independent, our IRSVM algorithm (the sequential version) will keep the independence property throughout the whole process, so that the least squares problem (5) has a unique solution $\beta^*$,

$$\beta^* = (\bar{K}'\bar{K})^{-1}\bar{K}'K(A, A_i'). \tag{7}$$

The distance $r$ from $K(A, A_i')$ to the column space of $\bar{K}$ is the squared root of the optimal value of (5) and is computed by

$$r = \left\|\bar{K}\beta^* - K(A, A_i')\right\|_2. \tag{8}$$

The square distance can be written in the form $r^2 = (I - P)K(A, A_i')$, where $P = \bar{K}(\bar{K}'\bar{K})^{-1}\bar{K}'$ is the projection matrix of $R^m$ onto the column space of $\bar{K}$. In other words, $r^2$ is the excess information carried in $K(A, A_i')$ over $K(A, \bar{A}')$. Note that the size of the reduced set is very small, hence it will not lead to any computational difficulty in solving the least squares problem, though we have to solve it many times in the whole process. Below we describe two algorithms, sequential and batch versions.

**Algorithm 3.1 IRSVM Algorithm (sequential version)**
Let $\delta > 0$ be a given threshold.

1) Choose a very small random subset matrix $\bar{A}_0 \in R^{\bar{m}\times n}$ from the training data matrix $A \in R^{m\times n}$, say $\bar{m} = 2$, as an initial reduced set, and generate the reduced kernel matrix $K(A, \bar{A}_0')$. Let $\bar{A}_{new} = \bar{A}_0$.
2) Select $A_j \in A \backslash \bar{A}_0$ and compute the distance $r$ from the kernel vector $K(A, \bar{A}_j')$ to the column space of $K(A, \bar{A}_{new}')$ by using (8).
3) If $r > \delta$ then $\bar{A}_{new} = \bar{A}_{new} \cup A_j$.
4) Repeat Step 2) until several successive failures happened in 3), then the resulting $K(A, \bar{A}_{new}')$ is our final reduced kernel.
5) Apply the Newton-Armijo Algorithm [9] to solve the objective function (1):

$$\min_{(\bar{u},\gamma)\in R^{\bar{m}+1}} \Phi_{\alpha,\nu}(\bar{u}, \gamma), \tag{9}$$

where the reduced kernel $K(A, \bar{A}')$ in (1) is that obtained in Step 4).
6) The separating surface is given as follows:

$$K(x', \bar{A}')\bar{D}\bar{u} = \gamma, \tag{10}$$

where $(\bar{u}, \gamma) \in R^{\bar{m}+1}$ is the unique solution to (9).
7) A new point $x \in R^n$ is classified into class $+1$ or $-1$ depending on whether the step function:

$$(K(x', \bar{A}')\bar{D}\bar{u} - \gamma)_*, \tag{11}$$

is $+1$ or zero, respectively.

We note that we have to solve normal equations (6) many times in Algorithm 3.1. The time complexity of this step is $O(\bar{m}^3)$, where $\bar{m}$ is the current size of the reduced set. In fact, the main cost of solving the normal equations depends on $\bar{K}'\bar{K}$, but not on $K(A, A_j')$. For example, if we have the $LU$ decomposition of $\bar{K}'\bar{K}$, we will get the solution of (6) by backward and forward substitution for any $K(A, A_j')$ [5]. In order to speed up Algorithm 3.1, we take the advantage of this fact, and use a batch of data points to generate the new reduced set. We propose a batch version IRSVM algorithm as follows:

**Algorithm 3.2 IRSVM Algorithm (batch version)**
Let $\delta > 0$ be a given threshold and $l$ be a given batch size. $\bar{B}_i \in R^{l\times n}$ denotes a batch of data and $r$ denotes a distance vector.

1) Choose a very small random subset matrix $\bar{A}_0 \in R^{\bar{m}\times n}$ from the training data matrix $A \in R^{m\times n}$, say $\bar{m} = 2$, as an initial reduced set, and generate the reduced kernel matrix $K(A, \bar{A}_0')$. Let $\bar{A}_{new} = \bar{A}_0$.
2) For $A_j$ to $A_{j+l} \in A \setminus \bar{A}_0$, form a batch $\bar{B}_i$.
3) For each $\bar{B}_i$, compute the distance vector $r$, which consists of individual distances from each of the columns of $K(A, \bar{B}_i')$ to the column space of $K(A, \bar{A}_{new}')$ by using (8).

4) For each $A_j \in \bar{B}_i$, $\bar{A}_{new} = \bar{A}_{new} \cup A_j$ if the corresponding distance value exceeds the threshold $\delta$.
5) Repeat Step 3) until several successive failures in adding new points. Then $K(A, \bar{A}'_{new})$ is our resulting reduced kernel.
6) Apply the Newton-Armijo Algorithm [9] to solve the objective function (1):

$$\min_{(\bar{u},\gamma)\in R^{\bar{m}+1}} \Phi_{\alpha,\nu}(\bar{u},\gamma), \qquad (12)$$

where the reduced kernel $K(A, \bar{A}')$ in (1) is that obtained in Step 5).
7) The separating surface is given as follows:

$$K(x', \bar{A}')\bar{D}\bar{u} = \gamma, \qquad (13)$$

where $(\bar{u}, \gamma) \in R^{\bar{m}+1}$ is the unique solution to (10), and $x \in R^n$ is a free input space variable of a new point.
8) A new point $x \in R^n$ is classified into class $+1$ or $-1$ depending on whether the step function:

$$(K(x', \bar{A}')\bar{D}\bar{u} - \gamma)_*, \qquad (14)$$

is $+1$ or zero, respectively.

This modified algorithm significantly reduces the number of times of solving the normal equations. One awareness is that "*similar*" points in the same batch might be added into our reduced set. But for the reason of speed, this algorithm is considerable. In next section, we will show experimental results about the performance of our algorithm.

## 4. Numerical Results and Comparisons

We applied IRSVM on four publicly available test problems, Cleveland Heart Problem, BUPA Liver, Ionosphere and Pima Indians from the University of California (UC) Irvine repository. The numerical results showed that IRSVM achieved comparable test set correctness with SSVM and RSVM, while with a much smaller reduced set size than RSVM.

We implemented the batch version of IRSVM using standard native MATLAB commands and used Gaussian kernel, $K(x, z') = e^{-\mu\|x-z\|_2^2}$, $x, z \in R^n$ for all our numerical tests. All parameters in these tests were chosen for optimal performance on a tuning set, a surrogate for a test set. All our experiments were run on a personal computer, which consists of Pentium-4 2.4GHz processor, 256 megabytes of memory and utilizing the Windows 2000 operating system.

In order to evaluate how well each method generalizes to future data, we performed ten-fold cross-validation on each dataset [15]. We also used *stratification* scheme in splitting the entire dataset to keep the "similarity" between training and testing datasets. That is, we try to make the distributions of training and testing sets as closed as possible [17]. A smaller testing error indicates a better prediction ability. The results are shown in Table I. It can be found that IRSVM used a smaller reduced set than RSVM to generate the nonlinear separating surface at a comparable accuracy level.

## 5. Conclusion

We have proposed an Incremental Reduced Support Vector Machine (IRSVM) that starts with an extremely small reduced set and then sequentially expands to include informative data points into the reduced set. These informative data points will be identified by solving a small least squares problem. Our approach provides a mechanism to determine the size of the reduced set automatically and dynamically but not pre-specified and the reduced set generated by this method will be more representative than the one by purely random selection. We have also proposed a batch version of IRSVM that will save some computation effort in solving a series of least squares problems. All advantages of RSVM for dealing with large scale nonlinear classification problem are retained. Moreover, the experimental tests on four publicly available dataset from the University of Irvine repository show that IRSVM used a smaller reduced set than RSVM without scarifying classification accuracy. With all these properties, IRSVM appears to be a very promising method for handling large scale classification problems using a nonlinear separating surface.

## References

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

| Tenfold Test Set Correctness % / Tenfold Computational Time, *Seconds* | | | | | |
|---|---|---|---|---|---|
| Dataset Size $m \times n$ | Methods | | | | |
| | IRSVM | | RSVM | | SSVM |
| | Correctness % / Time *sec* | $\bar{m}$ | Correctness % / Time *sec* | $\bar{m}$ | Correctness % / Time *sec* |
| Cleveland Heart | 85.53 | | 85.60 | | 85.59 |
| $297 \times 13$ | 4.70 | 17.6 | 2.04 | 30 | 23.20 |
| BUPA Liver | 73.97 | | 74.24 | | 73.65 |
| $345 \times 6$ | 7.59 | 18.3 | 2.75 | 35 | 30.59 |
| Ionosphere | 95.20 | | 95.17 | | 96.02 |
| $351 \times 34$ | 7.90 | 15.1 | 3.48 | 35 | 36.14 |
| Pima Indians | 76.84 | | 76.82 | | 76.69 |
| $351 \times 34$ | 13.40 | 14.8 | 9.07 | 35 | 168.90 |

TABLE I

TENFOLD CROSS-VALIDATION CORRECTNESS RESULTS ON FOUR UC IRVINE DATASETS. $\bar{m}$ VALUE IS THE TENFOLD AVERAGE OBTAINED BY RUNNING THE BATCH VERSION IRSVM.


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

[3] 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.

[4] G. Fung and O. L. Mangasarian. Proximal support vector machine classifiers. In F. Provost and R. Srikant, editors, *Proceedings KDD-2001: Knowledge Discovery and Data Mining, August 26-29, 2001, San Francisco, CA*, pages 77–86, New York, 2001. Asscociation for Computing Machinery. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-02.ps.

[5] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, Maryland, 3rd edition, 1996.

[6] Trevor Hastie Rob Tibshirani Ji Zhu, Saharon Rosset. 1-norm support vector machines. submitted to NIPS 2003.

[7] 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.

[8] Yuh-Jye Lee and Wen-Feng Hsieh. $\epsilon$-ssvr: A smooth support vector machine for $\epsilon$-insensitive regression. 2003. IEEE Transactions on Knowledge and Data Engineering (submitted).

[9] Yuh-Jye 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.

[10] O. L. Mangasarian. *Nonlinear Programming*. SIAM, Philadelphia, PA, 1994.

[11] 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.

[12] 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.

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

[14] P. M. Murphy and D. W. Aha. UCI machine learning repository, 1992. www.ics.uci.edu/∼mlearn/MLRepository.html.

[15] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, 36:111–147, 1974.

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

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