

Between-Subclass Piece-wise Linear Solutions in Large Scale Kernel SVM Learning

Tejas Indulal Dhamecha^a, Afzel Noore^b, Richa Singh^a, Mayank Vatsa^a

^a*IIT Delhi, India*

^b*Texas A&M University-Kingsville, USA*

Abstract

The paper proposes a novel approach for learning kernel Support Vector Machines (SVM) from large scale data with reduced computation time. The proposed approach, termed as Subclass Reduced Set SVM (SRS-SVM), utilizes the subclass structure of data to effectively estimate the candidate support vector set. Since the candidate support vector set cardinality is only a fraction of the training set cardinality, learning SVM from the former requires less time without significantly changing the decision boundary. SRS-SVM depends on a domain knowledge related input parameter, i.e. number of subclasses. To reduce the domain knowledge dependency and to make the approach less sensitive to the subclass parameter, we extend the proposed SRS-SVM to create a robust and improved hierarchical model termed as the Hierarchical Subclass Reduced Set SVM (HSRS-SVM). Since SRS-SVM and HSRS-SVM splits non-linear optimization problem into multiple (smaller) linear optimization problems, both of them are amenable to parallelization. The effectiveness of the proposed approaches is evaluated on four synthetic and six real-world datasets. The performance is also compared with traditional solver (LibSVM) and state-of-the-art approaches such as divide-and-conquer SVM, FastFood, and LLSVM. The experimental results demonstrate that the proposed approach achieves similar classification accuracies while requiring fewer folds of reduced computation time as compared to existing solvers. We further demonstrate the suitability and improved performance of the proposed HSRS-SVM with deep learning features for face recognition using Labeled Faces in the Wild (LFW) dataset.

Keywords: Support vector machines, subclass, subcluster, piece-wise linear solutions, large scale learning.

Email addresses: tejasd@iiitd.ac.in (Tejas Indulal Dhamecha), afzel.noore@tamuk.edu (Afzel Noore), rsingh@iiitd.ac.in (Richa Singh), mayank@iiitd.ac.in (Mayank Vatsa)

1. Introduction

Currently, the size of the largest biometric database is more than a billion, the number of individuals with bank accounts has increased by more than 700 million in the last three years, and YouTube generates billions of views everyday. The availability of similar large volumes of diverse data has lead to an ever-growing popularity and importance of machine learning and pattern classification algorithms, particularly scalable machine learning models. Traditionally, the most important parameter in selecting a classification model, for a given problem, has been the accuracy of the classifier ; however, due to rapid growth in the size of the databases, scalability of the classifier is now another important factor.

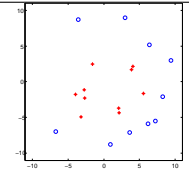
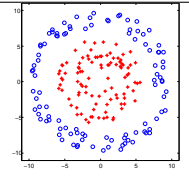
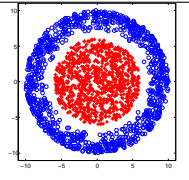
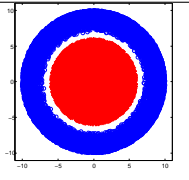
A large number of classification techniques exist in the machine learning literature; each with their own advantages and limitations. Support vector machine (SVM) [1] has been one of the widely used classification algorithms in a variety of domains and has shown excellent results in various applications including computer vision related problems (e.g. object classification [2], face recognition [3], and pedestrian detection [4]). The applicability and efficacy for real world applications has lead to numerous variants of SVM being proposed in the literature to make large scale training efficient [5, 6, 7, 8, 9]. However, there are two major limitations of SVM in context to large data:

- **Computational complexity:** The core optimization function of SVM is a quadratic programming (QP) problem. Therefore, the training time complexity of standard SVM is $O(n^3)$ [10], where n is the number of training instances.
- **Space complexity:** Training an SVM has space complexity of $O(n^2)$ [10]. This estimate is, typically, dominated by the space required for storing the kernel matrix.

To better understand the role of high time and space complexities, we show an example with a synthetic dataset termed as **two concentric circles (2CC)**. As illustrated in Table 1, it is a 2D two-class dataset where the samples of each concentric circular band corresponds to one class. Computing the nonlinear hyperplane to separate the two concentric circles requires learning kernel SVM models. As shown in Table 1¹ experiments are performed with varying number of data points in each circle. The results in Table 1 show that depending on the number of points in the two circular bands, the training time of SVM changes significantly. With 20 data points, LibSVM requires 1.6×10^{-2} seconds whereas, with 20,000 data points, i.e., increasing the number

¹The training time is computed with LibSVM.

Table 1: Training time as a function of the number of training instances for a synthetic two-dimensional dataset two concentric circles (2CC). Super-linear computational requirements of SVM training is evident from figures suggesting a need for efficient large scale kernel SVM.

Visualization				
Number of instances	20	200	2000	20,000
Training time (sec.)	1.6×10^{-2}	3.6×10^{-1}	1.9×10^1	3.3×10^3

by three orders of magnitude, the training time increases to five orders of magnitude. This shows that traditional SVM solvers are not optimized for large scale learning.

To learn a large scale kernel SVM, this paper introduces Subclass Reduced Set (SRS) SVM². It focuses on splitting the nonlinear optimization problem into multiple linear optimization sub-
 35 problems each operating on a significantly smaller fraction of the training data. Since the SVM solvers typically have super-linear time complexity, applying solver on such small sized candidate set yields significant time improvements. This proposed SRS-SVM leverages subclass³ structures of data in order to reduce the time complexity of obtaining the decision boundary. In order to compute the subclass structure of data, the proposed SRS-SVM relies on *number of subclasses*
 40 (parameter h). The benefit of SRS-SVM can be limited (time saving or accuracy) if h is inappropriately high (e.g. each sample is a subclass) or low (e.g. only one subclass). In real world datasets, it may be difficult to set a balanced value for h due to unknown distribution. To address this issue, we propose an extension that relaxes the dependency on exact parameter value and can operate with reasonably high value of h . This extension is manifested in a tree-like generalized
 45 hierarchical version of the proposed SRS-SVM termed as Hierarchical SRS (HSRS)-SVM. Due to the inherent property of solving small sub-problems, the proposed approaches are parallalizable and computationally fast, while maintaining the classifier accuracies. To show the effectiveness of SRS-SVM and HSRS-SVM, experiments are performed on four synthetic nonlinear datasets and

²Code available at <http://iab-rubric.org/resources/srs-svm.html>

³ In the literature, the term subclass [11, 12] is often used to describe a subset of samples of a class with certain shared characteristics. In a statistical learning sense, subclasses do not necessarily correspond to subclusters, as a subclass may contain multiple subcluster or vice-versa [13]. For example, on fitting Gaussian mixture model to a class, more than one mode may be considered as a single subclass if they are close enough to each other. However, to follow the vocabulary of other SVM specific related works [11, 12], the term *subclass* is used to mean subcluster.

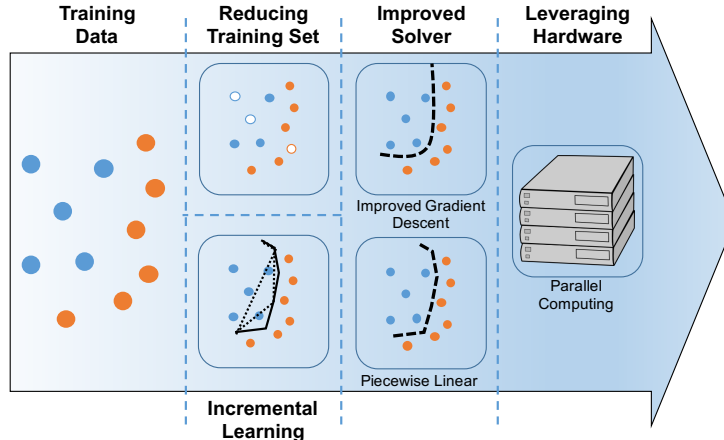


Figure 1: Illustrating four categories of approaches designed for scalable SVM learning.

six real-world datasets, namely `adult` [14], IJCNN1 [15], CIFAR-10 [16], forest cover (`covertype`)
 50 [17], face detection dataset from the Pascal Large Scale Learning Challenge (LSL-FD) [18], and
 Labeled Faces in the Wild (LFW) dataset [19]. The results are shown in comparison to LibSVM
 and state-of-the-art SVM variants proposed for handling large scale data.

2. Literature Review

The approaches proposed for scalable SVM can be grouped into four categories: (i) reduced
 55 training set size, (ii) incremental learning, (iii) improved solver, and (iv) leveraging hardware.
 As shown in Figure 1, these algorithms either operate at one of the steps involved in the SVM
 pipeline or incrementally update the learned model. We next review some of the algorithms in
 each of the four categories.

2.1. Reducing Training Set Size

60 The approaches that operate at data/input stage, generally, propose to reduce the size of the
 training set by either dividing or reducing the training set into subsets. Since all the subsets
 are operated independently, the process is inherently parallelizable. Moreover, these approaches
 operate at the first stage of training, therefore their benefits are observed into the subsequent
 solver and execution stages as well. From all the subsets, the required information is extracted,
 65 for instance, Lagrange multipliers and candidate support vectors. Later, the information from
 individual subsets is combined to obtain the final model.

Among one of the first such approaches, Yu et al. [20] proposed a top-down hierarchical
 clustering approach. At first samples are clustered, followed by clustering of the centroids; the

procedure is repeated recursively producing a hierarchical clustering. Intuitively, going from root
70 to leaf level clusters is akin to exposing the model to finer details of decision boundary step-
by-step. The initial model is learned from the centroids of the top (biggest) clusters. In the
subsequent stages, the model is updated based on the children (smaller) clusters. A similar top-
down approach is proposed by Boley and Cao [21]. Graf et al. [22] proposed Cascade SVM to
learn models on disjoint subsets of the training set in parallel. The final SVM model is learned
75 on a cumulative set of SVs obtained after iteratively processing the subsets.

Another set of techniques has focused on reducing the training set size in order to formulate
scalable SVM models. In the ideal case, the reduced training set should consist of only those
samples which are support vectors of the global solution. The reduced SVM and its variants
[23, 24] include a candidate SV set selection stage followed by learning standard SVM model on
80 it. Similarly, Ilayaraja et al. [25] aimed at estimating concise candidate SV set in the multi-class
scenario by exploiting the redundant nature of SVs amongst individual binary classifiers. Wang
et al. [26] explored the geometric interpretation of SVM to obtain the candidate SV set. Nath
and Shevade [27] utilized clustering based approach to eliminate parts of training data. Recently,
Hsieh et al. [28] proposed a divide-and-conquer SVM (DCSVM). Kernel k -means is first employed
85 to divide the training set into subsets and a set of support vectors (SV) is obtained from each
subset. The SVs are pooled and considered as the refined training set. Iteratively, the subsets are
created using kernel k -means and the models are learned. The number of subsets is reduced in
each subsequent iteration. DCSVM is currently one of the fastest SVM variants. Although not
with the focus on scalability, Tong and Koller [29] proposed an active learning based approach to
90 mitigate the need for large dataset.

2.2. Incremental Learning

A set of approaches inspired from incremental learning paradigm are also explored in the
literature. These approaches learn from incremental data streams and do not require to operate
on the whole training set. This inherently results in reduced space requirements. Incremental
95 SVM [30, 31] variants have been introduced for more than two decades now and have been utilized
for various applications including biometrics. Since incremental SVM approaches do not require
to keep the whole training set in the memory, their space complexity, typically is scalable for
large training sets. Syed et al. [30] empirically showed that to incrementally update an existing
SVM model, it is sufficient to learn a model from the combined pool of existing SVs and the
100 SVs of the incremental batch. As an offshoot, it provides an empirical basis for utilizing SVs as
the representative of the decision boundary. Ralaivola and dAlché Buc [32] proposed using the

locality information to update an SVM model with a new sample. Poggio and Cauwenberghs [33] provided a theoretical framework to increment or decrement the existing SVM model with a sample. Karasuyama and Takeuchi [34] extended the framework for incrementing existing SVM model with multiple samples. Recently, Mehrotra et al. [35] proposed a variant that explores the granular structure of data for efficient incremental learning.

2.3. Improved Solver

This category focuses on making the quadratic programming solver of SVM more efficient to handle large datasets. They can be grouped into either improving the gradient descent or obtaining the piece-wise linear solutions.

- **Improved Gradient Descent:** One of the earliest research for addressing the computationally highly complex constrained QP focuses on reformulating the objective function in an unconstrained optimization function [36]. The proposed least square SVM classifier operates on the primal formulation by reformulating the optimization function into a set of linear equations. Other research efforts in similar directions are by Shalev-Shwartz et al. [8], Bottou and Lin [37], and Langford et al. [38] that use iterative algorithms such as stochastic gradient descent. Although extremely efficient for learning linear SVMs, the major limitation is that the approaches in this category may be difficult to apply with kernel SVMs due to their primal formulations and/or large kernel matrix computations.
- **Piece-wise Linear Solutions:** These techniques operate by approximating the actual optimization problem. Such approaches focus on utilizing the intuition that even a nonlinear decision boundary is linear in small sections/local regions [12]. Huang et al. [39] proposed a piece-wise linear SVM approach via piece-wise linear feature mapping. Similarly, Fornoni et al. [40] proposed an approach that can leverage the piece-wise linear structure in the multiclass scenario with class specific weights. Ladicky and Torr [41] proposed to obtain local coding of each data point based on its local neighborhood. However, this approach is not aimed for large scale learning. Kecman and Brooks [42] proposed to use the training samples in the vicinity of a query sample to obtain the final classification. Recently, Johnson and Guestrin [43] modeled the piecewise linearity property in terms of working set selection for improved scalability. It is to be noted that the locally linear SVM variants are not necessarily developed with the focus on large scale learning. However, they provide the basis for utilizing the locally linear structure of complex decision boundaries for nonlinear classification.

2.4. Leveraging Hardware

135 This category is motivated by the availability of parallel computing hardware. The focus is to modify the solver algorithms for execution on multicore or multiprocessor environment. The research direction exploring the use of parallel processing and the hardware technology such as multicore processors [10] and distributed computing environments [44, 45] has resulted in various SVM variants for large scale learning. Zanni et al. [46] proposed parallelization of stochastic
140 gradient descent to exploit the multicore architecture of processors. Tsang et al. [10] proposed core vector machine that is specifically designed for utilizing multiple cores of processors. In order to efficiently leverage distributed and parallel processing environment, Do and Poulet [47] proposed a variant of least square SVM. Moreover, the inherently incremental nature of the approach makes its space complexity more suitable for large scale learning. Caragea et al. [48] and Forero
145 et al. [49] proposed approaches that rely on exchanging support vectors among sites (processing units) to learn the model in distributed computing environments. Do and Poulet [50] proposed to partition the training data and to learn parallel local SVM models on each of them. In a similar partitioning-based approach, Guo et al. [51] proposed to leverage map-reduce framework for training SVM in heterogeneous parallel computation infrastructure. Other approaches proposed
150 for efficient large scale learning include utilization of semi-supervised training data [52], leveraging the sparse nature of training data [53], and approximating the kernel equivalent high dimensional representation [54].

In view of the literature, the proposed Subclass Reduced Set (SRS) SVM is positioned at the intersection of subset-based and piece-wise linear approaches. By exploiting piece-wise linearity of
155 decision boundary between subclass-pairs, we divide the large scale kernel SVM learning problem into easier linear SVM learning problems. Further, these linear solutions form the basis for obtaining reduced training. Thus, the proposed research improves upon literature by proposing novel computationally efficient approaches by marrying the concepts of training set reduction and piece-wise linearity.

160 3. Preliminaries of SVM

This section briefly summarizes the basic formulation of support vector machine and defines some terms to facilitate explanation of the proposed approach.

SVM [1] is one of the widely used classification technique which falls under the category of discriminative classifiers. Let \mathbf{x}_i , $i = \{1, 2, \dots, n\}$ be n training samples and $y_i = \pm 1$ be
165 their corresponding class labels. A part of the objective of linear SVM is to obtain a projection

direction \mathbf{w} and a bias b such that samples of each class are on different sides of the separating plane. i.e. $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$. Further, practically useful formulation of SVM utilizes *soft margin* that tries to obtain as much cleaner decision boundary as possible by allowing misclassification of training samples to a certain degree (represented by slack variable ξ_i), i.e. $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i$.
 170 Correspondingly, the optimization problem takes the form of

$$\arg \min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \quad \text{s.t.} \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad (1)$$

where, C is the misclassification cost. Eq. 1 is called the *primal* form of the (soft-margin) SVM optimization function. By utilizing the Lagrangian multipliers α , the equivalent *dual form* of the optimization becomes

$$\arg \max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j, \quad \text{s.t.} \quad 0 < \alpha_i \leq C \quad (2)$$

Having obtained the optimal multipliers α , the projection direction \mathbf{w} is obtained as, $\mathbf{w} =$
 175 $\sum_i \alpha_i y_i \mathbf{x}_i$. As \mathbf{w} and b define the hyperplane separating samples from two classes, and that \mathbf{w} is defined as a linear summation of \mathbf{x}_i makes it intuitive that only the α_i corresponding to the samples near the decision boundary are non-zero. Only these samples with non-zero multipliers, that contribute in defining \mathbf{w} , are called Support Vectors (SVs). All the points that are outside the margin get zero coefficient value assigned. In other words, $\alpha_i = 0$, $i \in \{j | y_j(\mathbf{w} \cdot \mathbf{x}_j + b) > 1\}$.

180

4. Reduced Set and Variants

We present the definitions and propositions associated to reduced set with respect to SVMs.

Definition 1. *Reduced Set (RS)* is a subset of the training set indices. For a training set with n samples, the index set $T_{RS} \subset \{1, 2, \dots, n\}$ defines a Reduced Set.

185 **Definition 2.** *Representative Reduced Set (RRS)* is a reduced set that yields the same decision boundary as the whole training set. Let α and $\hat{\alpha}$ represent the Lagrangian coefficients for the optimization functions of the whole training set and its reduced set T_{RS} , respectively. T_{RS} is a representative reduced set if $\sum_{i=1}^n \alpha_i y_i \mathbf{x}_i = \sum_{j \in T_{RS}} \hat{\alpha}_j y_j \mathbf{x}_j$.

Definition 3. *Minimal Representative Reduced Set (MRRS)* is the smallest possible RRS.
 190 T_{MRRS} is an MRRS of the train set if there exists no other RRS with less cardinality than T_{MRRS} .

Proposition 1. *Representative Reduced Set (RRS) contains all the support vector indices.*

Proof. Let T_{SV} and T_{nSV} be the index sets of support vectors and non-support vector samples, respectively. The direction \mathbf{w} can be written as,

$$w = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i = \sum_{j \in T_{SV}} \alpha_j y_j \mathbf{x}_j + \sum_{k \in T_{nSV}} \alpha_k y_k \mathbf{x}_k \quad (3)$$

Since, $\forall k \in T_{nSV}$, $\alpha_k = 0$, $w = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i = \sum_{j \in U} \alpha_j y_j \mathbf{x}_j$, such that $T_{SV} \subset U$ and $U \subset \{1, 2, \dots, n\}$.

195 Therefore, every RRS (set U) contains all the support vector indices, i.e. $T_{SV} \subset T_{RRS}$. \square

Proposition 2. *Minimal Representative Reduced Set (MRRS) contains only the support vector indices and maximum cardinality of MRRS is $|T_{SV}|$.*

Proof. From Proposition 1, $T_{SV} \subset T_{RRS}$.

The reduced representative set T_{RRS} can further be written as $T_{RRS} = T_{SV} \cup M$, where M contains only
200 the non-support vector indices, i.e. $M \subset T_{nSV}$.

Since the non-support vectors have no impact on the value of \mathbf{w} , all of them can be discarded to reduce the cardinality of T_{RRS} .

Therefore, if a T_{RRS} is an MRRS, at most, it can contain all the support vector indices and no other indices; i.e. $|T_{MRRS}| \leq |T_{SV}|$. \square

205 Based on these definitions and propositions, it can be inferred that 1) RRS would contain *all* the support vector indices and 2) MRRS would contain only the support vector indices. In the proposed approach, we focus on obtaining the best possible estimate of MRRS in order to reduce the computational time without affecting the classifier performance.

5. Proposed Subclass Reduced Set SVM

210 Proposition 1 implies that if we can estimate the candidate SV set, it can be utilized to obtain the same decision boundary as obtained from the whole train set. If the estimated candidate set contains m samples and $m \ll n$, then the optimization function can be solved with reduced computation and space requirements. In other words, the training time can be reduced significantly, as (1) the number of support vectors is typically very small compared to the total number
215 of training samples, i.e. ($|T_{SV}| \ll n$) and (2) the SVM solvers, typically, have quadratic time complexity. Further, leveraging this property is well suited in large datasets, as the inequality $|T_{SV}| \ll n$ is held strongly in densely sampled datasets. Based on this premise, we propose an approach, termed as Subclass Reduced Set SVM (SRS-SVM), to learn SVM with lower training complexity compared to a traditional solver. As illustrated in Figures 2 and 3, the proposed SRS-
220 SVM has two stages: (1) estimating the MRRS ($|T_{MRRS}| \ll n$) and (2) solving the optimization

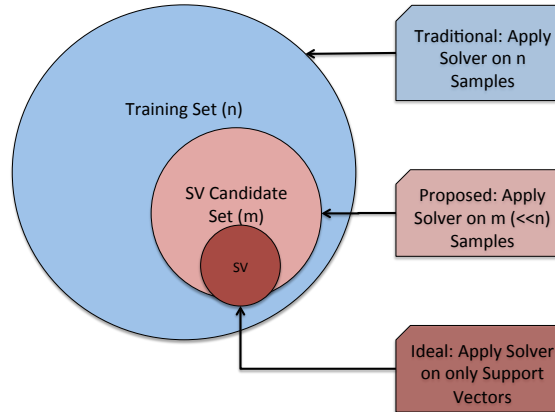


Figure 2: Abstract illustration explaining the core concept of the proposed approach, Subclass Reduced Set SVM. Approaches, such as SRS-SVM, that fall under the categorizations of the subset based and piece-wise linear approaches, operate on this basic intuition.

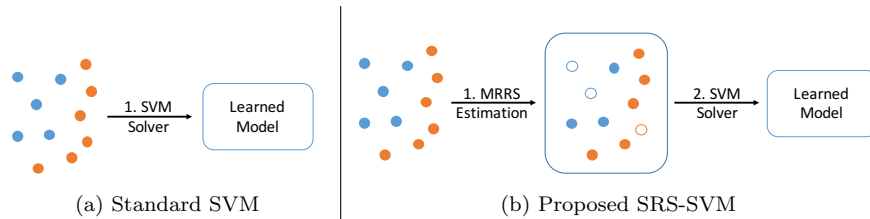


Figure 3: Traditionally SVM solver is applied on the complete training set. The proposed SRS-SVM operates in two stages: estimating MRRS and applying SVM solver on the obtained reduced set. For a detailed illustration of MRRS estimation block, refer to Figure 4.

function on the estimated MRRS. Stage-2 requires less training time as opposed to solving the optimization function on the whole training set; however, a significant training time improvement is achievable only if MRRS is estimated efficiently in Stage-1. Therefore, the proposed approach relies on the efficient estimation of MRRS in order to reduce the overall computational cost.

225 *5.1. Estimating Minimal Representative Reduced Set*

The detailed concept of the proposed subclass reduced set SVM is illustrated in Figure 4. We use piece-wise linearity of nonlinear solutions and the subclass structure of data for estimating MRRS. Details of the MRRS estimation approach are explained below.

5.1.1. Leveraging subclass structure of data

230 It is well understood that real-world data may form subclasses within a class [20, 11]. Samples sharing some common property may create a subclass within a class. Since, the variation between subclasses is smaller than the variation between classes, subclasses may provide a fine-grained

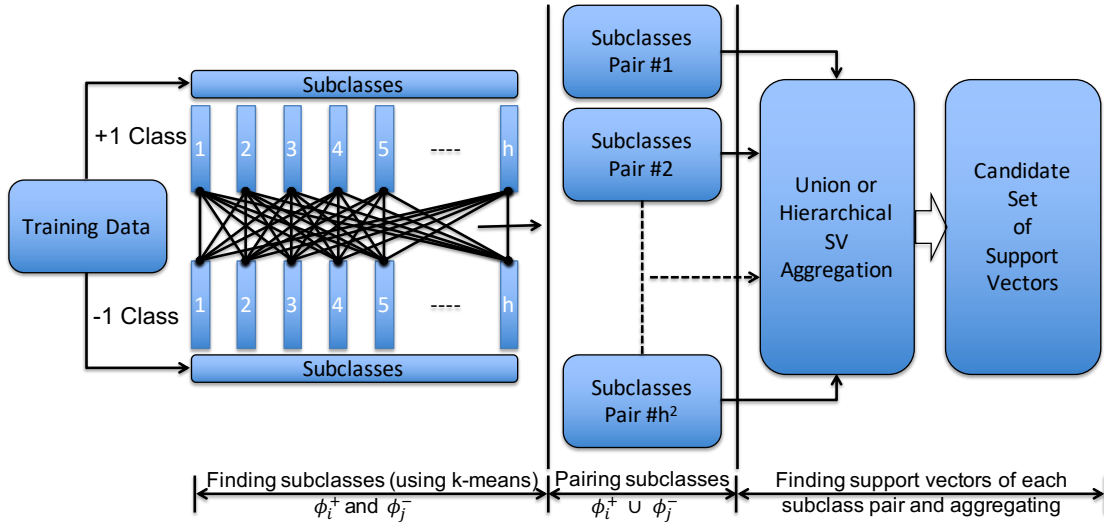


Figure 4: Block diagram of MRRS estimation procedure of the proposed Subclass Reduced Set SVM. Each class is divided into h subclasses. (Here, subclasses are obtained using k-means clustering) Each subclass of $+1$ class is paired with each subclass of -1 class, thus resulting in a total of h^2 subclass-pairs. Support vectors from each subclass-pair are retained as the candidate global support vectors. They are combined either using union operator (in SRS-SVM) or using a further hierarchical aggregation (in Hierarchical SRS-SVM).

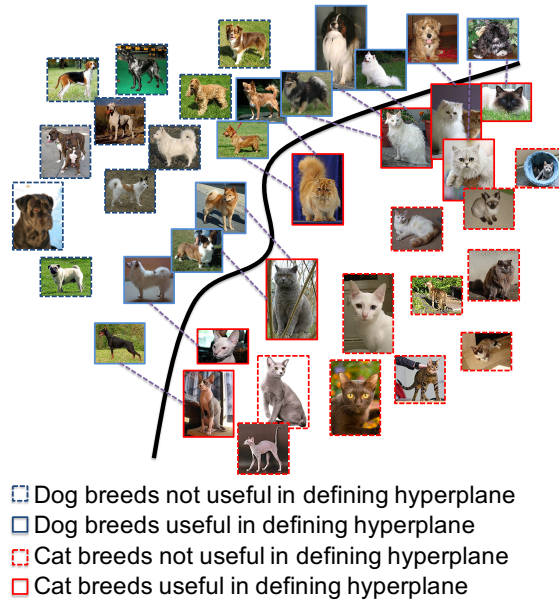


Figure 5: Illustrating the applicability of subclass structure in modeling decision boundary for Dog vs Cat classification problem. Out of a vast variety of dog and cat breeds, there are only limited breeds (subclasses) that contribute to the decision boundary. Localized decision boundaries between breed-pairs of dogs and cats can be seen as constituents for the overall decision boundary.

information of the data distribution within a class. Let us consider an example of Dog vs Cat classification problem as shown in Fig 5. There are certain ways in which dogs differ from cats, however, there are certain ways in which one breed of dog (e.g. German Shepherd) would differ from another breed of dog (e.g. Doberman Pinscher). In this example, dogs and cats represent classes whereas various breeds represent the subclasses. Researchers have attempted to exploit the notion of subclasses for different classifiers [12, 11, 55]. In this research, we explore the subclass notion for fast estimation of MRRS.

Clustering to find subclasses: As illustrated in Figure 5 subclasses represent a finer categorization of a class based on some shared characteristics (in this case, breed of dog). However, the subclass labels are typically not available, therefore, we have to estimate the (pseudo) subclass labels. As each subclass encompasses samples sharing some characteristics, naturally, its estimation is a clustering problem.

K-means clustering and approximation: Subclasses can be obtained with existing approaches such as k-means. Although more sophisticated approaches such as Gaussian mixture models [56], DBSCAN [57], and agglomerative clustering [58, 59] may be applied, we have observed that simple k-means suffices to efficiently estimate MRRS in the proposed framework. Note that exact solution of k-means clustering has a computational complexity of $O(n^{dh+1})$, which is usually inappropriate to be used practically. To address this, we use Lloyds algorithm, which provides a heuristic solution [60]. Since k-means using Lloyds algorithm is an iterative approach, by restricting the maximum number of iterations, the subclasses can be obtained in significantly less time. With this, the subclasses can be obtained with $O(n^+ dhp)$ and $O(n^- dhp)$ time complexities for class +1 and -1, respectively. n^+ , n^- , d , h , and p represent the number of samples in +1 class, the number of samples in -1 class, feature dimensionality, the number of subclasses, and the number of iterations, respectively.

5.1.2. Piece-wise linear solution to a nonlinear problem

It has been suggested in the literature that a nonlinear decision boundary can be achieved with the help of several piece-wise linear solutions (PWL) [39, 61, 62]. This notion also suggests that every piece-wise solution encodes discriminative characteristics of a slice of dataset lying in its vicinity. Since the decision boundaries are described using support vectors, it implies that the SVs obtained for each local region, jointly, can represent the overall nonlinear decision boundary. Thus, the SVs of piece-wise linear solutions can be utilized to estimate the representative reduced set. It is important to accurately define the local regions for obtaining the linear solutions and

subclass structure of the data can be leveraged for this purpose. Proposition 3 shows that local regions defined as the subclass-pair can be useful in obtaining the global nonlinear solutions.

Proposition 3. *If a sample is a support vector in the global nonlinear solution, it is a support vector in at least one of the subclass pair-wise solutions.*

270 *Proof.* If a sample x_i is a support vector in the global nonlinear solution, it is within the margin of the solution. Therefore, the sample x_p is on the boundary (hull) of its class. [63]

Let x_q be its nearest support vector in the opposite class ($y_p \neq y_q$). Since x_q is also a support vector, it is on the boundary (hull) of its class.

Without loss of generality, we can assume that x_p and x_q belong to i^{th} and j^{th} subclasses respectively, i.e. $p \in \phi_i^+$ and $q \in \phi_j^-$.
275

Therefore, x_p is on the boundary (hull) of the i^{th} subclass of +1 class and x_q is on the boundary (hull) of the j^{th} subclass of -1 class, and

x_p is a Support Vector in the solution learned for the subset $\phi = \phi_i^+ \cup \phi_j^-$ □

Proposition 3 brings together the notion of piece-wise linear solutions and the subclass structure of data by providing the basis for utilizing the subclass structure to obtain the PWL solutions
280 for MRRS estimation. The PWL solutions make it possible to obtain pairs of subclasses that can be utilized to obtain support vectors. Let π be an indicator variable such that $\pi(x_i)$ denotes the subclass association of the i^{th} sample. Let both the classes be divided into h subclasses each⁴, $\phi_i^+ = \{k | \pi(x_k) = i \ \& \ y_k = +1\}$ represents the index set of samples of +1 class belonging to the
285 i^{th} subclass, and, similarly, $\phi_j^- = \{k | \pi(x_k) = j \ \& \ y_k = -1\}$ represents the index set of samples of -1 class belonging to the j^{th} subclass, where $i, j \in \{1, 2, 3, \dots, h\}$. Decision boundaries obtained for the pairs $\phi_i^+ \cup \phi_j^-$ describe a set of possible hyperplanes discriminating two classes in local regions. All the h^2 pairs of subclasses can be utilized for obtaining the global solution. Estimating minimal representative reduced set requires solving h^2 sub-problems defined on subclass-pairs. A
290 degenerate case for this is when each sample is considered as a subclass where each subclass-pair solver is bound to yield a linear decision boundary. With approximately reliable subclass association, each subclass-pair decision boundary can be assumed to be linear. Overall, estimation of MRRS involves learning h^2 linear solvers and aggregating their SVs. For simplicity, we assume that each subclass of +1 and -1 classes have $\frac{n^+}{h}$ and $\frac{n^-}{h}$ samples respectively. As a result, at
295 first, h^2 linear SVM models are learned; each of which is learned over $((n^+ + n^-)/h)$ samples.

⁴For the ease of mathematics, we assume that both the classes are divided into equal number of subclasses. However, that is not a constraint of the proposed SRS-SVM.

Algorithm 1 Proposed Subclass Reduced Set SVM

procedure

Input: Data matrix X , number of subclasses h , cost C , and kernel hyper-parameters

▷ Find subclass association of each sample

$\pi = \text{findSubclasses}(X)$

▷ $\pi(x_i)$ denotes the subclass association of i^{th} sample.

▷ using k-means clustering.

$T_{RRS} = \{\}$

▷ Initialize reduced representative set

for $i = 1$ to h **do**

for $j = 1$ to h **do**

$\phi_i^+ = \{k | \pi(x_k) = i \ \& \ y_k = +1\}$

▷ index set of i^{th} subclass samples of +1 class

$\phi_j^- = \{k | \pi(x_k) = j \ \& \ y_k = -1\}$

▷ Index set of j^{th} subclass samples of -1 class

$\phi = \phi_i^+ \cup \phi_j^-$

▷ Index set for the subproblem

Solve the subproblem:

$\arg \max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j, \quad \text{s.t.} \quad 0 < \alpha_i \leq C, \quad i, j \in \phi$

$T_{RRS} = T_{RRS} \cup \{k | \alpha_k > 0\}$

end for

end for

▷ Learn final SVM model

Solve the nonlinear classification problem on the candidate support vector set

$\arg \max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j), \quad \text{s.t.} \quad 0 < \alpha_i \leq C, \quad i, j \in T_{RRS}$

Return: Learned SVM model on T_{RRS}

end procedure

Note that, this also removes the requirement of storing the whole $(n^+ + n^-) \times (n^+ + n^-)$ kernel matrix in the memory which is a bottleneck for large scale SVM learning.

Under the assumption of representative subclass categorization and appropriate parameterization, the union set of SVs corresponding to subclass-pair solutions is a representative reduced set. Note that the union set may not necessarily be an MRRS as the mechanism does not prevent a global non-support vector from getting introduced into the union set. In the best case scenario, when no global non-support vector is introduced in the union set, the obtained union set is MRRS, resulting in optimally minimal computation time and space requirements. Thus, as detailed in Proposition 3, this union set of SVs can be considered an approximate reduced representative set. The final classifier model is obtained by learning kernel SVM on this reduced set. Test samples are classified using this kernel SVM model in the traditional manner, i.e.

$$y_z = \text{sign} \left(\left[\sum_{i \in T_{RRS}} \alpha_i y_i k(x_i, z) \right] - b \right) \quad (4)$$

where, z is the test sample, y_z is its predicted label, and k is the kernel matrix. Algorithm 1 outlines the steps involved in the proposed subclass reduced set SVM.

Effect of Number of Subclasses h : Consider the most degenerate case, where each class is divided into as many subclasses as the number of samples ($n/2$). In this case, each subproblem operates on two samples - one from each class. Both the samples are bound to become support

Table 2: The effect of the number of subclasses on the size of estimated MRRS.

	Decreasing number of subclasses \rightarrow				
Subclasses (h)	$\frac{n}{2}$	$\frac{n}{2} - \Delta$	\dots	$h^* + \Delta$	h^*
Size of estimated MRRS	n	$\sim n$	$< n$	$\ll n$	$\ll n$

vectors, effectively passing all the training samples into the RRS. Although it is a valid RRS, it is not a good approximation of MRRS. This degenerate case represents the worst case scenario, where the obtained candidate set is same as the whole training set. Further, as shown in Table 2, any large value ($\sim \frac{n}{2}$) for h is likely to result in unsuitably very large MRRS set. At the opposite case, consider a scenario where the whole class is considered as one subclass, i.e. $h = 1$. This configuration is also not useful, as it will violate the assumptions regarding the piece-wise linearity defined on local regions. In summary, both, overestimation and underestimation of h , are likely to yield sub-optimal results, due to large candidate SV set or basic violation of piece-wise linearity assumptions, respectively.

As the number of subclasses h is varied from $n/2$ (maximum number of subclasses) to h^* (optimum number subclasses), the size of estimated MRRS varies between n and a value close to a total number of global support vectors ($\sim |T_S V|$). The optimal h^* depends on the geometric arrangement of the data; e.g. for XOR dataset $h^* = 2$ due to the presence of two distinct clusters for each class. However, for real-world high dimensional datasets, it is crucial to find a reasonably balanced estimate of h . The following Section proposes a solution to address this challenge.

5.2. Hierarchical Subclass Reduced Set SVM (HSRS-SVM)

The solution to the problem is either to estimate h^* or to devise a mechanism that can handle arbitrary higher value of h . Estimating h^* essentially reduces down to understanding the distribution of the class, similar to that in a generative modeling. In literature, estimation of subclasses is approached from various perspectives, such as supervised statistical criterion [64, 11] and unsupervised estimation of mixture modes [65, 66] often relying on expectation-maximization [67]. However, since, the philosophical foundations of SVM are in discriminative modeling, we avoid the route of estimating h^* . We focus on creating an extended approach that can provide relatively efficient model even with sub-optimal h . The improved extended approach is a hierarchical version of the proposed approach SRS-SVM. It gains robustness to over-estimation of h by filtering out global non-support vectors at multiple levels of hierarchy.

As shown in Figure 6, the mechanism of proposed HSRS-SVM can be described in a tree

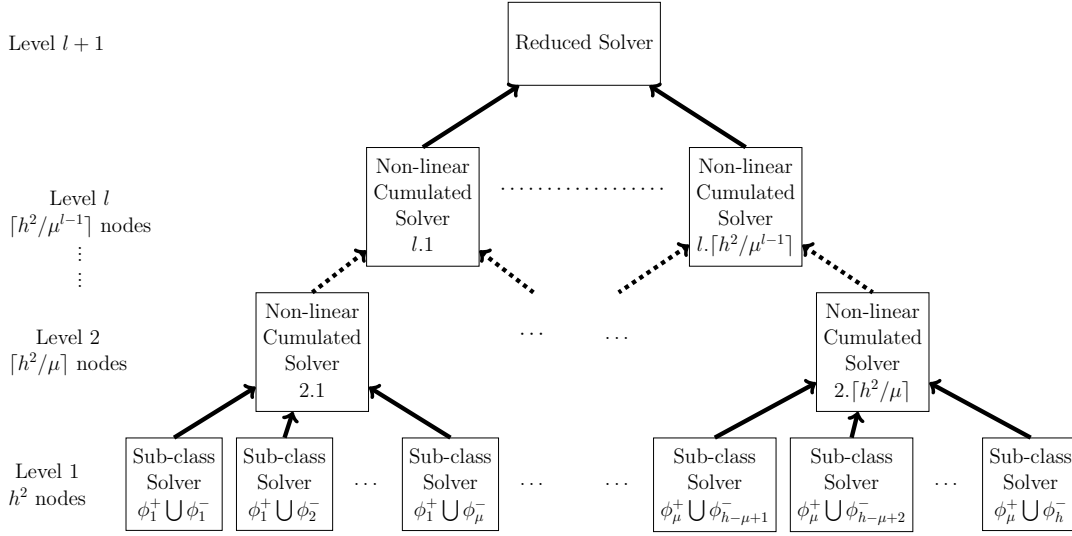


Figure 6: Graphical illustration of the proposed Hierarchical Subclass Reduced Set SVM (HSRS-SVM).

structure. Since the proposed algorithm follows bottom-up approach, our convention considers the leaf nodes at level 1. Each leaf node caters to one subclass-pair solver $\phi_i^+ \cup \phi_j^-$, i.e. a linear SVM is learned within each leaf node. Only the support vectors from each individual solvers are moved further up in the tree and the remaining samples are discarded. Further, a set of μ models is selected to learn an aggregated solver at the level 2. If each class is divided into h subclasses, there will be h^2 leaf nodes. In this work, a total of $\lceil h^2/\mu \rceil$ aggregated nodes are obtained at level 2. Based on the learned $\lceil h^2/\mu \rceil$ models, a total of $\lceil h^2/\mu^2 \rceil$ models are obtained at level 3. In general, the proposed approach operates on $\lceil h^2/\mu^{l-1} \rceil$ nodes at level l . The iterative aggregation stops at the root level which represents the final aggregated solver model. Since, μ nodes are aggregated at each level, the root node is placed at level k such that $\lceil h^2/\mu^{k-1} \rceil = 1$. Further, in the case of $\mu = h^2$, the root level itself becomes level 2, making the mechanism equivalent to SRS-SVM. Thus, the proposed *SRS-SVM* is a special case of *HSRS-SVM*.

To increase the chances of introducing samples from various parts of feature space into the next level, nodes are randomly shuffled prior to aggregation. This helps maintain representativeness of the data distribution at next level nodes. For example, without shuffling, the node 2.1 (in Figure 6) receives the support vectors from $\phi_1^+ \cup \phi_1^-$, $\phi_1^+ \cup \phi_2^-$, $\phi_1^+ \cup \phi_3^-$, ..., $\phi_1^+ \cup \phi_\mu^-$ subclass-pairs. All these subclass-pairs have one common (or repetitively occurring) subclass. The support vectors from these pairs provide a limited view of the overall data spread, as they only encode decision boundary between ϕ_1^+ and the parts of -1 class. Instead, if the nodes are shuffled, a relatively holistic nature of decision boundary may be encoded in the subsequent layers.

We can learn all the leaf nodes in parallel, as each node corresponds to training a separate
 360 linear SVM model. Thus, the total time for leaf level computation is, in the best case scenario,
 equal to the maximum time required for an individual solver. Further, the level 2 nodes can also
 be learned in parallel in a similar way. Thus, the total time required for the overall computation
 is $\sum_{i=1}^{l+1} \max(t_i^1, t_i^2, \dots)$, where t_i^j is the time required for training j^{th} node in i^{th} level. In practice,
 propagating the SVs upwards in the tree will also consume computational cycles; however, it will
 365 be negligible relative to learning SVM models in each node.

6. Datasets and Protocols

The effectiveness of the proposed SRS-SVM and HSRS-SVM is evaluated on both non-linearly
 separable synthetic datasets and real-world datasets. Datasets are chosen with considerable
 variations in characteristics such as feature dimensionality, training set size, and application
 370 domain (finance, weather, object images, face images, textual data) to show the applicability and
 efficacy of the proposed algorithm.

6.1. Nonlinearly Separable Synthetic Datasets:

The synthetic datasets enable performance evaluation in the presence of known nonlinearity
 characteristics. All the synthetic datasets are chosen to be two-dimensional, as they provide an
 375 opportunity to visualize the data scatter and the decision boundary.

1. Two concentric circles (2CC)
2. Three concentric circles (or bullseye)(3CC)
3. Shooting range (a set of bullseyes) (SR)
4. XOR dataset

Figure 7 illustrates the distributions of the above mentioned synthetic datasets utilized in this
 380 research. All the synthetic datasets are created by defining the distribution functions. Thus,
 we can arbitrarily sample varying number of instances from these datasets. For each dataset,
 experiments are performed with instance size varying between 100 to 100,00. Further, the datasets
 have a varying degree of nonlinearity. For example, the nonlinear nature of the databases increases
 385 as we proceed from two concentric circles dataset (2CC) to three concentric circles dataset (3CC)
 and then to the shooting range (SR) dataset.

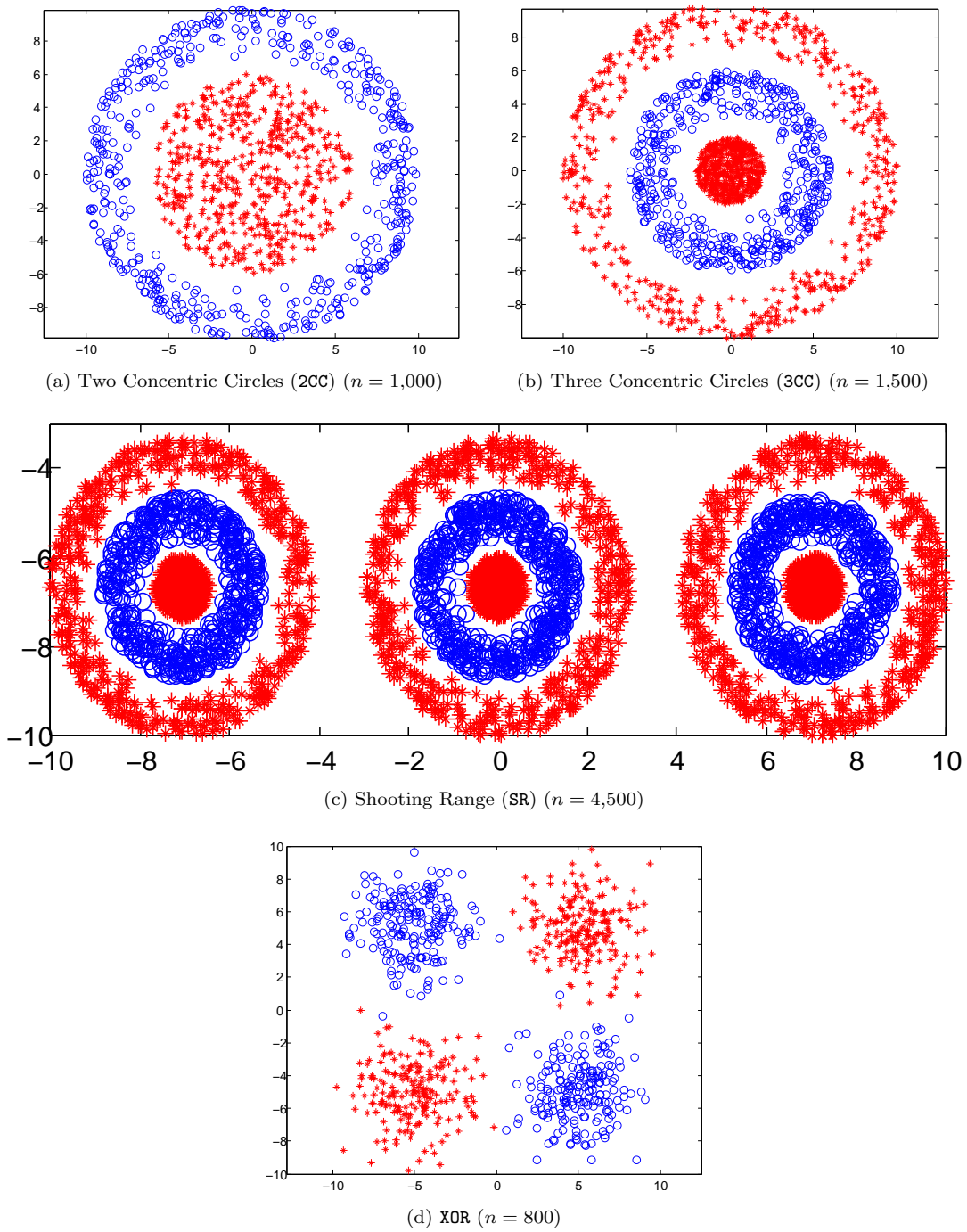


Figure 7: Illustrating the synthetic datasets used for performance evaluation (best viewed in color).

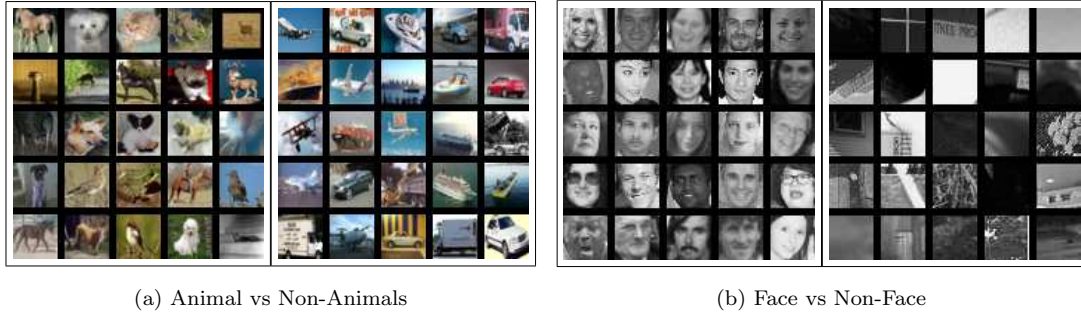


Figure 8: Samples of the real world databases used for performance evaluation: (a) animal and non-animal class images from face detection dataset of Pascal Large Scale Learning Challenge [16, 28] and (b) face and non-face images from face detection dataset of Pascal Large Scale Learning Challenge [18]

6.2. Real-world datasets

The proposed HSRS-SVM approach is evaluated on various real-world datasets. The datasets correspond to classification tasks in different fields of data analytics. The dataset characteristics are described in Table 3.

1. `adult/census income` [14]⁵: predicts whether a person’s income exceeds \$50K based on various demographic features from census data.
2. `ijcnn1` [15]⁶: consists of time-series of multiple observations from an internal combustion engine, with the goal of predicting normal and misfiring of the engine.
3. `covertype` [17]⁷: consists of cartographic measures of wilderness areas belonging to seven major forest cover classes. In this work, the dataset is converted to a binary class problem with the goal of separating class 2 from the remaining 6 classes (Protocol used in Collobert et al. [68]).
4. `cifar-10` [16]⁸: is an object detection dataset consisting of images of 10 object categories. However, in this work the categories are modified to classify between animals and non-animals (Protocol used in Hsieh et al. [28]). Figure 8(a) shows sample images from both the categories. Table 3 reports the parameter γ for the dataset is 2^{-22} (or 2.4×10^{-7}) which is considerably smaller compared to other datasets. As Hsieh et al. [28] suggest, the average distance between samples for un-scaled images of the dataset is much larger than

⁵<https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#a9a>

⁶<https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#ijcnn1>

⁷<https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#covertype.binary>

⁸<https://www.cs.toronto.edu/~kriz/cifar.html>

Table 3: Details pertaining to the real-world datasets used in the evaluation and their corresponding hyperparameters. (d is feature dimensionality, h is number of subclasses, C is misclassification cost, and γ is radial basis function kernel parameter)

Dataset (size)	No. of training samples	No. of testing samples	d	Parameters		
				h	C	γ
<code>adult</code> (45.8 MB)	32,561	16,281	123	15	1	2^{-5}
<code>ijcnn1</code> (23.78 MB)	49,990	91,701	22	5	2^5	2
<code>covertypes.binary</code> (239.36 MB)	464,810	116,202	54	500	4	2^5
<code>cifar-10.binary</code> (1.37 GB)	50,000	10,000	3,072	30	2	2^{-22}
<code>LSL-FD</code> (1.34 GB)	150,000	50,000	900	50	10	1

405 other datasets, which requires smaller values of the γ as the multiplicative factor to sample distance term $\|\mathbf{x}_i - \mathbf{x}_j\|$.

5. Face detection from Pascal Large Scale Learning Challenge (`LSL-FD`) [18]: the dataset consists of a large number of face and non-face images. It is useful for benchmarking face detection performance. Figure 8(b) shows sample face and non-face images.

410 For `adult`, `ijcnn1`, and `cifar-10` datasets, the predefined benchmarking train-test splits, available at the source, are utilized. An 80-20% train-test split of the binarized `covertypes` is used based on a protocol used in literature [68]. For the `LSL-FD` dataset, 75-25% train-test split of 200K samples is utilized in the experiments.

7. Experiments on Synthetic Datasets

415 In the first part of the evaluation, we use synthetic datasets to understand the effectiveness of the proposed approach. As the proposed approach relies on an approximation of original objective functions, the decision boundaries obtained with SRS-SVM are compared with a traditional solver (LibSVM).

7.1. Visualization of Each Step

420 We first demonstrate the functioning of the proposed SRS-SVM by providing the visualization of various stages of the algorithm on `XOR` dataset. The scatter plot of training samples is shown in Figure 9(a). The next step involves processing the sub-class pairs with $h = 2$. Figure 9(b) shows $h^2 = 4$ subclass-pairs along with a linear SVM decision boundary obtained from each of the subclass-pair based subproblems. All the linear decision boundaries along with the scatter

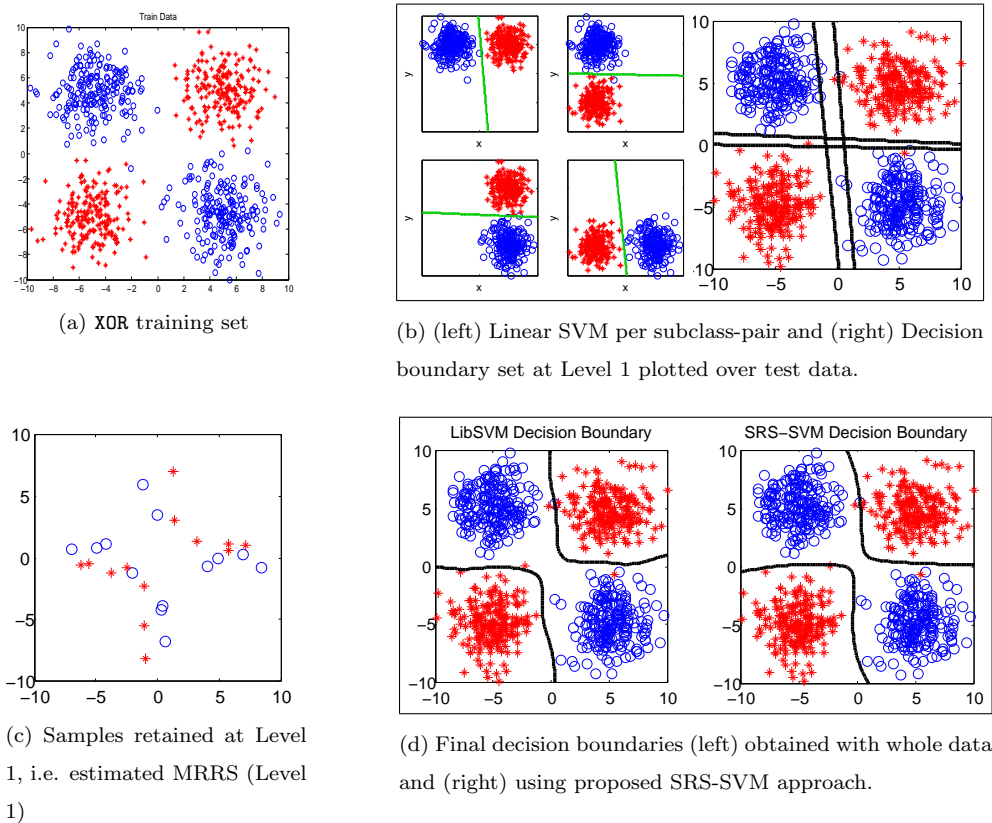


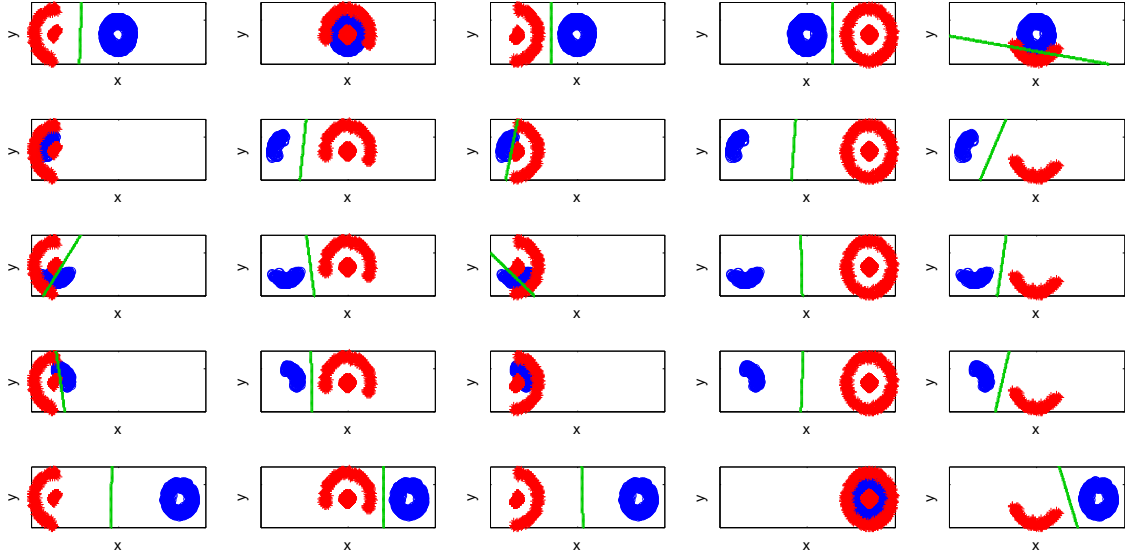
Figure 9: Visualization of proposed approach on the XOR dataset. Training on whole dataset ($n = 800, h = 2$) LibSVM takes 3.46 seconds; whereas the proposed SRS-SVM obtains similar decision boundary in 0.25 seconds. See Algorithm 1 to relate the mathematical formulation of the individual steps.

425 plot of estimated MRRS (candidate SV set) is shown in Figure 9(c). Out of $n = 800$ training samples, only 26 are retained as candidate SV set. Thus, a large fraction (96.7%) of samples are discarded at this stage. The final classification boundary obtained using the proposed SRS-SVM is shown in Figure 9(d) (right). Comparing this with the decision boundaries obtained by applying LibSVM on the entire training set shows that both the decision boundaries are very

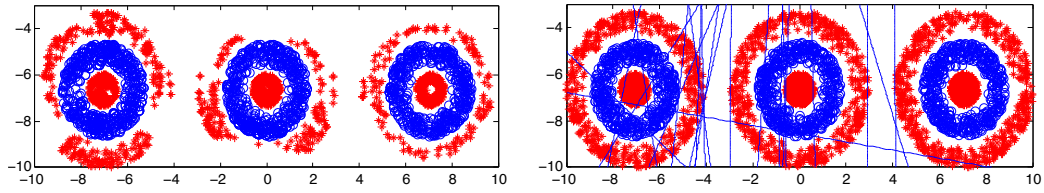
430 similar for the classification task.

Figure 10 shows the working of the proposed SRS-SVM algorithm on the SR (Shooting range) dataset. As the number of subclasses (h) is parametrized to 5, the linear decision boundary is learned for 25 subclass-pairs. It can be observed that a large portion of samples from the outermost band are rejected. The samples lying on the outer boundary of the band are not in the vicinity of the margin of separation, which leads to their rejection as shown in Figure 10(c).

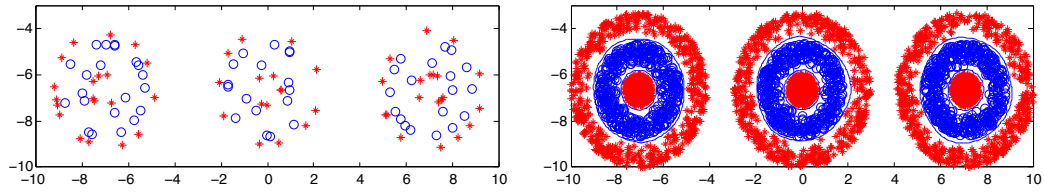
435 At the end of Level 1, approximately 3,609 samples are retained out of the total 4,500 training



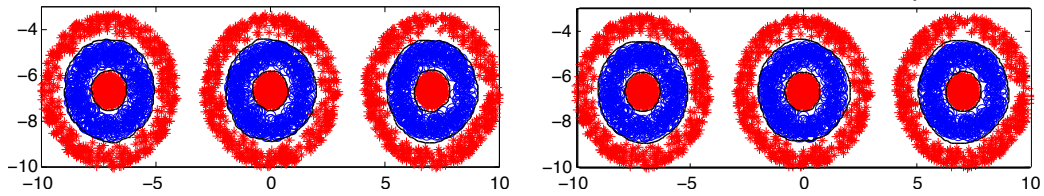
(a) One linear SVM decision boundary is learned for each of the 25 subclass-pairs obtained by dividing each class into 5 subclasses.



(b) (left) Samples retained as candidate SVs at Level 1 (leaf nodes). (right) Corresponding decision boundaries at Level 1.



(c) (left) SVs at Level 2 (root node). (right) Corresponding decision boundary at Level 2.



(d) Final decision boundaries (left) obtained with whole data and (right) using proposed SRS-SVM.

Figure 10: Illustrating the processing of the proposed SRS-SVM on the Shooting Range dataset (see Fig. 7c). Training on the whole dataset ($n = 4,500$) LibSVM takes 93 seconds; whereas the proposed SRS-SVM obtains similar decision boundary in 50 seconds.

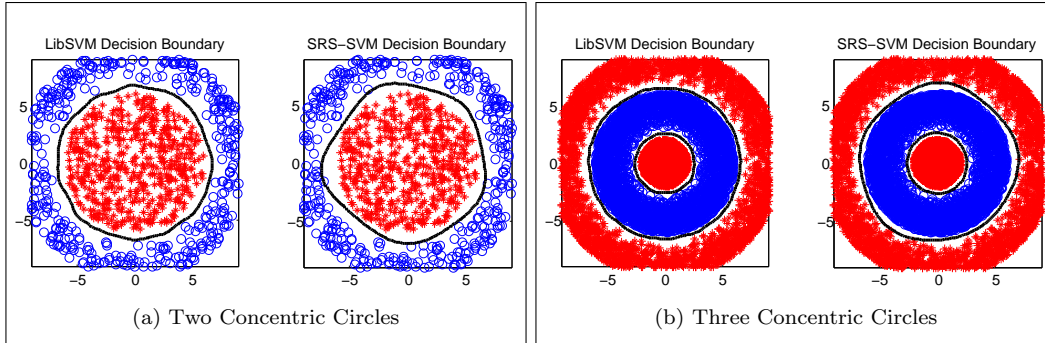


Figure 11: Comparative illustration of the decision boundaries obtained by LibSVM and by the proposed SRS-SVM approach ($h = 5$).

samples. The SR dataset does not have clearly visible five subclasses; however, due to the mechanism of learning h^2 linear SVMs, the proposed approach yields the decision boundary similar to that obtained with LibSVM. With minimal reduced representative set (MRRS) estimation, the proposed approach is able to reduce the training time by almost half as compared to LibSVM. Similarly, the decision boundary comparison for the other two synthetic datasets, is shown in Figure 11. The XOR dataset actually contains two subclasses, the class corresponding to inner circle of 2CC has actually only one subclass (the class itself), and for 3CC and SR datasets it is hard to concretely define the number of subclasses due to their nonlinearity. However, while applying SRS-SVM, we set the number of subclasses $h = 5$ for all these datasets. Although, it is an inexact parameterization, in all the cases, the decision boundaries obtained with the proposed SRS-SVM are almost same as (visually) those obtained with LibSVM. The efficacy of SRS-SVM with inexact parameterization helps understand its performance in application areas with limited domain knowledge.

7.2. Quantitative Analysis

In order to understand the time improvement of the SRS-SVM, we generate varying number of samples (between 100 and 10000) from each synthetic dataset. The training time of the proposed approach and LibSVM is compared as a function of the number of training samples. Figure 12 shows the graphs corresponding to this experiment for 2CC, 3CC, and XOR datasets. Figure 13 shows similar graphs for the SR (shooting range) dataset, with results for additional analysis pertaining to the number of subclass parameter (h).

For all the datasets, both SRS-SVM and LibSVM yield perfect classification on the test sets. The reported training time in this experiment includes the time required for estimating parameters C (misclassification cost) and γ using grid search, and the time required for training

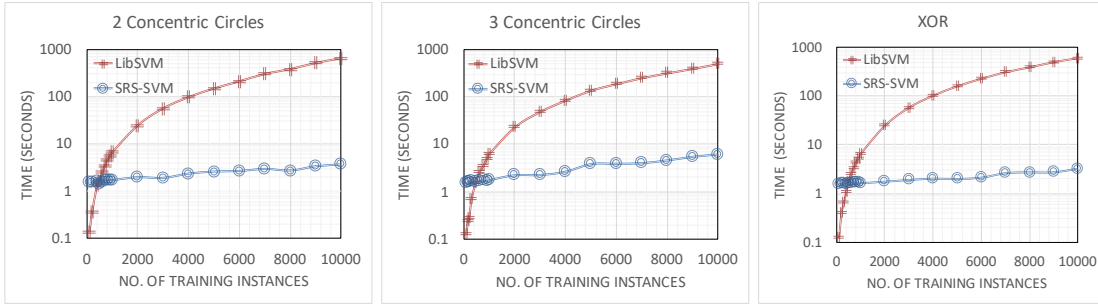


Figure 12: (Best viewed in color) Comparing training time on three synthetic datasets: two concentric circles (2CC), three concentric circles (3CC), and XOR. A varying number of samples are generated for each of the datasets. The training time is shown on the logarithmic scale. As the number of training instances increases, the training time of LibSVM increases rapidly whereas, the proposed SRS-SVM has a significantly lower rate of increase in training time.

460 the model. It can be observed in Figure 12 that for a training size above a certain limit (> 500) the training time of the exact solver (LibSVM) increases rapidly; whereas the rate of increase in the training time is very small in the case of the proposed SRS-SVM. For example, in the case of 2CC dataset with 10,000 samples, the proposed approach requires few seconds ($< 10s$) whereas, the exact solver requires few hundreds of seconds ($< 1,000s$) for learning a model. Figure 13 shows

465 similar quantitative analysis for Shooting Range dataset. Given that the dataset is relatively complex, we observe that increasing the number of subclasses from 5 to 20, reduces the training time, as it aids in significantly reducing the training set size. For example, for 9,000 training samples, training time required for LibSVM is 430.7s; whereas for SRS-SVM with $h = 5, 15, 20$ requires training time of 176.6s, 71.9s, and 64.3s, leading to the speedup of 2.43x, 5.99x, and

470 6.69x, respectively. Note that there is not really a trade-off of accuracy, as all configurations of SRS-SVM and LibSVM yield perfect classification on the test sets.

8. Experiments on Real-world Datasets

Experiments on diverse real-world datasets are also performed to study (1) the comparative performance of the proposed subclass reduced set based approach, (2) the computational time required at various stages of applying SRS-SVM (namely, clustering, level-1, and level-2), (3)

475 the effectiveness of the proposed representative reduced set (RRS) estimation procedure, and (4) to study the effect of parameters h (number of subclasses) and μ (number of children) on training time and classification accuracy. The first three objectives involve experiments to study the effectiveness of the proposed subclass reduced set based approach with a parameterization of

480 $\mu = h^2$ (and therefore, two levels of hierarchy) as detailed in Section 5.1. The experiment is further

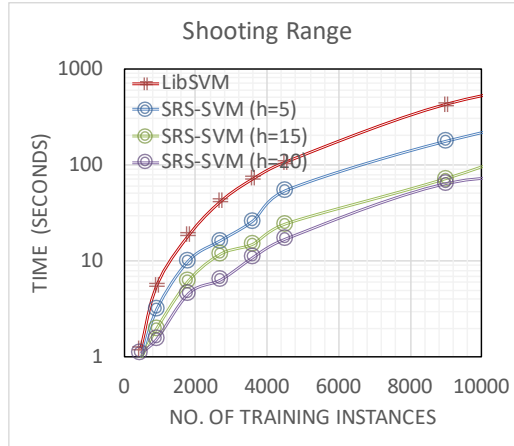


Figure 13: (Best viewed in color) Comparing training time on SR (Shooting Range) dataset. Different number of samples are generated from the dataset and training set size vs training time plots is shown for different dataset sizes with number of subclasses (h) as 5, 15, and 20. Consistently, SRS-SVM takes less training time compared to LibSVM. As the parameter h is increased, the training time is observed to reduce significantly on the logarithmic scale.

extended to the proposed hierarchical subclass reduced set SVM (HSRS-SVM) as described in Section 5.2.

8.1. Comparative Analysis

Comparison of the proposed subclass reduced set based approach with existing algorithm is performed with publicly available implementations. Hsieh et al. [28] have shown that large scale SVM approaches, namely Cascade SVM [22], SpSVM [53], and core vector machines [10] yield lower accuracies than DCSVM. Therefore, in this work, the results are compared with the most recent approaches namely DCSVM, LLSVM, and FastFood.⁹

1. LibSVM [69]: LibSVM is one of the widely used implementations of SVM that relies on sequential minimal optimization algorithm [14] for optimizing the QP objective function.
2. Divide and Conquer SVM (DCSVM) [28]: DC-SVM is one of the recent related approaches. In this study, the exact version of DC-SVM is utilized.
3. Low-rank Linearization SVM (LLSVM) [70]: We utilize the LLSVM implementation from the BudgetedSM toolbox [71].

⁹As the proposed approach relies on accurately finding a subset of the training set, it is logical to investigate the performance of a randomly sampled subset of training set. However, [28] have shown that such random subsets yield suboptimal performance.

Table 4: Comparing the results of the proposed HSRS-SVM to other related approaches.

(a) Classification Accuracy (%) comparison

Dataset	LibSVM [69]	LLSVM [70]	FastFood [54]	DCSVM [28]	Proposed ($\mu = h^2$)
adult	85.01	66.28	85.2	84.75	84.46
ijcnn1	98.70	98.34	91.58	98.39	97.82
covertypes.binary	96.07	71.25	out of memory	95.81	93.99
cifar-10.binary	89.66	78.27	79.79	89.78	89.92
LSL-FD	99.10	92.27	57.36	99.20	98.50

(b) Training Time (seconds) comparison

Dataset	LibSVM [69]	LLSVM [70]	FastFood [54]	DCSVM [28]	Proposed ($\mu = h^2$)
adult	135.4	99.4	83.1	122.6	60.2
ijcnn1	68.3	96.6	107.3	74.0	13.3
covertypes.binary	102,940.0	1,854.0	out of memory	75,183.0	47,536.0
cifar-10.binary	69,128.0	1,220.0	459.4	78,107.0	38,243.0
LSL-FD	311,543.0	1,396.5	254.0	515,674.0	112,558.0

495 4. FastFood [54]: The technique aims at obtaining approximate high dimensional representation.

Details regarding datasets and the hyper-parameters are provided in Table 3. The first set of experiments is performed with parameter $\mu = h^2$, which is a special non-hierarchical case of HSRS-SVM. The results of the comparative prediction performance and training time requirement are reported in Table 4(a) and Table 4(b), respectively. All the experiments are performed on a Windows machine with two 2.66 GHz Intel Xeon E5640 processors with 48GB primary memory. Table 4(b) (on page 26) shows that compared to LibSVM, the proposed algorithm yields, the speedup of 2.25x (135.4/60.2), 5.13x (68.3/13.3), 2.16x (102,940/47,536), 1.80x (69,128/38,243), and 2.76x (311,543/112,558) on **adult**, **ijcnn1**, **covertypes**, **cifar-10**, and **LSL-FD**, respectively, while yielding similar classification accuracies. Moreover, the speedup of 2.03x, 5.56x, 1.58x, 2.04x, and 4.58x with respect to DCSVM is observed in the case of **adult**, **ijcnn1**, **covertypes**, **cifar-10**, and **LSL-FD**, respectively. The basic assumption of the proposed approach is that estimating the candidate support vector set beforehand helps reduce the overall time complexity. The speedup compared to exact solver can be achieved only if the time consumed in estimating

510 the candidate support vector set is lesser than the time saved in learning the SVM model from it. If the dataset is densely sampled, the size of the candidate set is typically a small fraction of the whole training set; almost, guaranteeing improvement in speed. Typically, an exact model learned from a densely sampled set has a relatively very small number of support vectors (e.g. `ijcnn1`, `adult`, and LSL-FD) which leads to a significant speed-up with the proposed subclass
 515 reduced set based approach. To further compare the proposed HSRS-SVM and LibSVM, we perform McNemar’s test to evaluate if marginal homogeneity exists between the predictions of the two. The test reveals that for `adult` ($p = 0.0058$) and `ijcnn-1` ($p < 0.001$) datasets, the difference is statistically significant, whereas for `cifar` ($p = 0.08$) the difference is not statistically significant.

520 8.2. Training Time of Individual Stage

To further understand the proposed HSRS-SVM approach, we provide its stage-wise training times in Table 5. As explained earlier the first stage involves obtaining subclasses, which is followed by Level 1 of training involving the estimation of MRRS based on h^2 linear SVM decision boundaries, and Level 2 involves learning nonlinear decision boundary. Training time of each stage
 525 is reported on absolute and relative scale. It is observed that the subclass computation stage takes a very small fraction (0.2-10%) of the total training time. This is a very supportive result as any computationally heavy subclass computation stage can affect the overall computation for large scale learning. These results also imply that utilizing more time-efficient subclass computation approach may not result in further reducing the training time significantly. Level 1 computation
 530 involving MRRS estimation consumes a 6 – 49% of training time. However, this stage involves learning of h^2 linear SVMs independently, thus using parallel architecture (e.g. multi-threading) can further reduce the computation time of Level 1 by multiple folds. Overall, we observe that the Level 2 (i.e. learning nonlinear SVM on estimated MRRS) requires more than 50% of the total training time due to the complex nature of kernel SVM learning.

535 8.3. Effectiveness of MRRS Estimation Approach

This analysis is presented to understand how effectively the proposed subclass based approach estimates the reduced representative set. In order to understand this, its precision and recall are computed with respect to the support vector set (T_{SV}) of the exact solver. If an estimated MRRS (\hat{T}_{MRRS}) is a minimal RRS (i.e. smallest possible RRS), it will overlap completely with
 540 T_{SV} . Moreover, for an estimated MRRS to have as less spurious candidate support vectors, its *precision*, computed as $\frac{|\hat{T}_{MRRS} \cap T_{SV}|}{|\hat{T}_{MRRS}|}$, should be close to one. Similarly, for an estimated MRRS

Table 5: Stage-wise training time of the proposed subclass reduced set based SVM approach. Time is reported in seconds. The figures in the parenthesis represent the fraction of total training time consumed in percentage. Level 2 is the root level as $\mu = h^2$.

Dataset	Subclass computation	Level 1 (MRRS estimation)	Level 2 (Learning decision boundary from estimated MRRS)
adult	3.5 (5.8%)	14.8 (24.6%)	41.9 (69.6%)
ijcnn1	1.2 (9.1%)	5.3 (40.2%)	6.7 (50.7%)
covtype.binary	235.9 (0.5%)	2,978.3 (6.3%)	44,315.6 (93.2%)
cifar-10.binary	269.5 (0.7%)	5,855.9 (14.7%)	33,635.4 (84.6%)
LSL-FD	228.7 (0.2%)	42,693.0 (38.1%)	69,636.0 (61.7%)

to have all the actual support vectors, its *recall*, computed as $\frac{|\hat{T}_{MRRS} \cap T_{SV}|}{|T_{SV}|}$, should be close to one.

The precision and recall for the set of SVs in the final SVM model of the proposed approach ($T_{r,SV}$) is also computed. The metrics help in quantifying the similarity between the SVM model of the exact solver and that obtained with the proposed HSRS-SVM. Note that, this quantification of similarity of two models is independent of the test set. Table 6 summarizes the results pertaining to this particular analysis. Key observations are as follows:

- As a general trend it can be observed that recall of estimated MRRS \hat{T}_{MRRS} is high ($> 80\%$) for all the datasets (except LSL-FD). This means the proposed MRRS estimation approach retains a large fraction of actual support vectors.
- The basic premise of the MRRS estimation is that it should retain *all* support vectors, i.e. recall is one. The recall of < 1 results from the following two practical aspects: 1) estimating subclasses using a limited iteration approximate k -means without actually modeling the data distribution, and 2) approximating the potentially nonlinear decision boundary of subclass-pairs with a linear decision boundary. Note that both of these approximations yield a significant improvement in training time, with recall > 0.8 . Table 4 shows that the trade-off does not have a significant impact on the classification accuracy.
- The precision of the MRRS estimation shows that majority of its elements are actual support vectors. A close-to-one precision is not necessary to obtain SVM model equivalent to the traditional solver. However, higher precision of RRS estimate reduces the training time of

Table 6: Numerical analysis of the precision and recall of the estimated minimal reduced representative set (\hat{T}_{MRRS}) and the final support vector set (T_{rSV}) obtained using proposed HSRS-SVM approach with respect to the support vector set (T_{SV}) of the traditional solver (LibSVM).

Dataset	$ T_{SV} $	\hat{T}_{MRRS} (Estimated MRRS)			T_{rSV}		
		$ \hat{T}_{MRRS} $	Precision	Recall	$ T_{rSV} $	Precision	Recall
			$\frac{ \hat{T}_{MRRS} \cap T_{SV} }{ \hat{T}_{MRRS} }$	$\frac{ \hat{T}_{MRRS} \cap T_{SV} }{ T_{SV} }$		$\frac{ T_{rSV} \cap T_{SV} }{ T_{rSV} }$	$\frac{ T_{rSV} \cap T_{SV} }{ T_{SV} }$
adult	11,622	13,698	0.7220	0.8509	9,889	0.9093	0.8403
ijcnn1	2,478	10,865	0.1913	0.8390	2,202	0.8629	0.8390
covertypes.binary	98,978	242,998	0.3475	0.8531	88,892	0.8685	0.7800
cifar-10.binary	31,750	36,842	0.7197	0.8351	26,616	0.9614	0.8060
LSL-FD	130,117	69,991	0.9536	0.5129	67,034	0.9956	0.5129

subsequent levels.

- The precision values of T_{rSV} is typically higher than that of T_{RRS} . This validates the hypothesis that the spurious support vectors in the reduced representative set get discarded in the subsequent levels. Theoretically, the recall of T_{rSV} cannot be higher than that of \hat{T}_{MRRS} , as $T_{rSV} \subseteq \hat{T}_{MRRS}$ (therefore, $\frac{|T_{rSV} \cap T_{SV}|}{|T_{SV}|} \leq \frac{|\hat{T}_{MRRS} \cap T_{SV}|}{|T_{SV}|}$).
- In the case of LSL-FD dataset, estimated MRRS (\hat{T}_{MRRS}) is about half the size of the actual support vector set (T_{SV}). On other datasets, the estimated MRRS is larger than the actual support vector set. Due to this peculiar behavior, we observe that recall values for LSL-FD are lower as compared to other datasets. In spite of these observations, the classification performance is affected by only 0.6%, i.e. 99.1% by LibSVM vs 98.5% by the proposed subclass reduced set based approach in Table 4.

8.4. Effect of h (Number of Subclasses) and μ (Number of Children) Parameters in Hierarchical SRS-SVM

This experiment focuses on understanding the effect of the parameter h (number of subclasses) and μ (number of children) on training time and testing accuracy of the proposed HSRS-SVM. Table 2 outlines a theoretical relationship between number of subclasses (h) and size of the estimated MRRS ($|\hat{T}_{MRRS}|$). As explained earlier, a large value of h can render the time improvements ineffective, whereas a very small value can affect the performance. As detailed in Section 5.2, HSRS-SVM can relax the need of fine tuning h by introducing hierarchical structure to the MRRS estimation. The proposed hierarchical structure, which is controlled by μ (number of children),

Table 7: Effect of varying number of subclasses (h) and number of children (μ) on the training time and classification accuracy of the proposed HSRS-SVM on the `adult` dataset. The training time is reported in seconds. The figures within parenthesis represent the classification accuracy.

Number of Subclasses (h)	Training Time in seconds (Accuracy in %)					
	$\mu = h^2$	$\mu = \lceil \frac{h^2}{2} \rceil$	$\mu = \lceil \frac{h^2}{4} \rceil$	$\mu = \lceil \frac{h^2}{8} \rceil$	$\mu = \lceil \frac{h^2}{16} \rceil$	$\mu = \lceil \frac{h^2}{32} \rceil$
2	88.0 (68.5)	112.8 (65.8)	n/a			
4	79.0 (82.0)	84.8 (80.7)	84.1 (82.9)	116.0 (77.1)	n/a	
6	68.4 (84.2)	84.2 (83.2)	80.8 (83.5)	103.1 (74.9)	115.8 (75.7)	122.2 (81.7)
8	68.9 (83.7)	98.8 (84.1)	81.2 (83.7)	70.7 (83.9)	94.4 (83.2)	120.2 (78.7)
10	68.1 (83.5)	101.9 (83.2)	82.9 (84.3)	78.2 (82.3)	98.2 (84.2)	116.0 (84.0)
15	70.9 (84.1)	93.2 (84.6)	95.8 (84.3)	80.0 (84.1)	80.2 (84.2)	94.1 (83.5)
20	77.8 (84.7)	111.0 (84.3)	98.0 (84.0)	87.9 (84.7)	82.1 (84.4)	103.0 (84.8)
25	81.4 (84.4)	112.6 (84.7)	106.6 (84.7)	94.7 (84.2)	86.1 (84.7)	112.6 (84.4)
30	88.2 (84.4)	125.0 (84.8)	114.8 (84.9)	107.3 (84.9)	92.4 (84.5)	123.6 (84.5)
35	91.2 (84.7)	132.7 (84.6)	130.0 (84.8)	119.0 (84.7)	102.7 (84.7)	97.7 (84.9)
40	95.5 (84.6)	145.2 (84.8)	139.1 (85.0)	127.6 (84.8)	111.8 (85.0)	100.0 (84.9)
45	102.4 (84.6)	153.1 (84.7)	145.0 (84.6)	138.0 (84.6)	123.7 (84.8)	110.1 (84.7)
50	106.6 (84.9)	162.7 (84.9)	161.0 (84.8)	147.2 (84.8)	133.2 (84.7)	119.4 (84.7)

Table 8: Effect of varying number of subclasses (h) and number of children (μ) on the training time and classification accuracy of the proposed HSRS-SVM on the `ijcnn1` dataset. The training time is reported in seconds. The figures within parenthesis represent the classification accuracy.

Number of Number of subclasses (h)	Training Time in seconds (Accuracy in %)					
	Number of children (μ)					
	h^2	$h^2/2$	$h^2/4$	$h^2/8$	$h^2/16$	$h^2/32$
2	50.8 (91.4)	50.6 (91.8)	n/a			
4	33.4 (92.0)	34.3 (95.3)	32.6 (91.4)	31.9 (93.4)	n/a	
6	25.2 (95.7)	23.2 (94.7)	34 (90.9)	21.3 (95.3)	20.7 (94.4)	21.8 (95.1)
8	17.9 (95.6)	19.1 (94.9)	19.0 (95.8)	15.6 (95.1)	16.3 (94.7)	16.7 (95.8)
10	15.4 (96.4)	17.0 (95.6)	13.3 (95.4)	9.1 (96.2)	13.8 (96.2)	13.3 (95.1)
15	12.9 (96.8)	13.5 (96.2)	12.4 (96.9)	10.0 (96.1)	8.8 (96.6)	10.9 (96.2)
20	13.4 (97.4)	13.9 (97.5)	11.9 (96.9)	11.5 (97.3)	10.5 (97.3)	11.2 (96.9)
25	15.7 (97.4)	16.0 (97.6)	14.4 (97.6)	13.7 (97.9)	12.3 (97.7)	13.3 (97.7)
30	17.5 (97.8)	19.1 (98.0)	16.8 (98.0)	15.3 (97.4)	13.8 (97.8)	14.7 (97.4)
35	19.1 (98.1)	21.2 (98.1)	19.2 (97.6)	18.7 (98.3)	17.1 (98.2)	15.8 (97.8)
40	20.6 (98.3)	22.9 (98.2)	20.9 (98.2)	22.1 (97.9)	18.3 (97.9)	17.9 (98.1)
45	22.8 (98.0)	23.6 (98.2)	24.6 (98.3)	24.3 (98.2)	21 (98.1)	19.6 (98.2)
50	25.3 (98.3)	27.5 (98.3)	26.8 (98.4)	25.1 (98.0)	24.3 (98.4)	24.1 (98.2)

should yield good results with an approximate parameterization of h . This experiment focuses on verifying the expected behavior of the proposed hierarchical SRS-SVM. The number of subclasses h is varied between 2 and 50. For every value of h , experiments are performed with six different values of μ (h^2 , $h^2/2$, $h^2/4$, $h^2/8$, $h^2/16$, and $h^2/32$). Since μ has to be a natural number, a ceiling value is used.

Tables 7 and 8 summarize the results for `adult` and `ijcnn1` datasets respectively¹⁰. Similar trends were observed on other datasets as well. Note that, $\mu < h^2/2$ with $h = 2$, and $\mu < h^2/8$ with $h = 4$ are invalid combinations (mentioned as n/a) as they do not satisfy the condition $\mu \geq 1$.

- In our experiments, we observe that as the number of subclasses increase, the training time decreases around moderate value (~ 15 subclasses) and then increases steadily. The testing accuracy appears to increase rapidly but the rate of increase decreases at higher h approaching saturation. Note that as h increases, so does the size of estimated MRRS which is likely to reduce approximation explaining the accuracy convergence.
- When h is very small, the estimation of MRRS can be poor, i.e. it has low recall (many actual support vectors may be missed) and/or low precision (many non-support vectors are retained). The former will lead to poor testing accuracy, whereas the later will increase the computation time of subsequent levels by increasing the overhead of discarding non-SVs. It can be verified from Table 7 that underestimation of h results in overall poor testing accuracy and suboptimal training time.
- Similarly, higher values of h increases the size of estimated MRRS, which affects its precision and overall the training time adversely. However, it improves the recall of MRRS estimation, resulting in the convergence of the decision boundary and testing accuracy to that of an exact solver. As shown in Table 7 on `adult` dataset, the classification performance appears to converge/saturate at $h \geq 20$.
- In our experiments, we observe that, for constant h , varying μ from h^2 to $h^2/32$ increases the overall training time because for smaller μ we need to learn more number of intermediate models. For example, with $\mu = h^2$, h^2 linear SVMs (at Level 1) and 1 nonlinear SVM (at root Level 2) is learned internally; whereas, with $\mu = h^2/2$, h^2 linear SVMs at Level 1, 2 nonlinear SVMs at Level 2, and 1 nonlinear SVM at root Level 3 is computed. This effect is more pronounced with small values of h , as they lead to relatively higher number of samples per subclass; which make the training computationally expensive. However, with higher values of h it is still suitable to set μ at lower values, which can increase prediction performance with relatively less impact on overall training time.

¹⁰Due to the exhaustive nature of this experiment, we show tabular results on only two datasets.

Table 9: Verification accuracy of L-CSSE features with HSRS-SVM ($h = 5, \mu = 25$) and LibSVM in comparison to state-of-the-art approaches on the LFW database.

Approach	Accuracy
L-CSSE with HSRS-SVM	90.92
L-CSSE with LibSVM [72]	90.51
L-CSSE with Neural Net [72]	90.49
Spartans [73]	87.55
POP-PEP [74]	91.10
MRF-Fusion-CSKDA [75]	95.89

8.5. HSRS-SVM with Deep Learning Features for Face Recognition

To further investigate the performance and suitability of the proposed classifier we perform experiments on a challenging problem of face verification. In the last few years, deep learning based approaches have established state-of-the-art results in various research areas, especially in computer vision and face recognition. These approaches benefit from utilizing deep learning based features as inputs to traditional classifiers. Therefore, it is our assertion that the proposed subclass reduced set based SVM may also efficiently utilize deep learning based features. Further, this integration of deep learning feature with HSRS-SVM is expected to achieve improved accuracy (by virtue of the features) and to be computationally efficient (by the virtue of the proposed classifier). For face verification, we use Labeled faces in the wild (LFW) dataset [19]. The dataset consists of face images with the objective of face verification i.e. predicting match and non-match pairs. The face verification performance is reported for image-restricted protocol. The official protocol defines 10 fold cross-validation splits over 3000 match and 3000 non-match pairs. Each cross-validation contains 5400 images for training and 600 images for testing. We explore the utility of Local Class Sparsity Based Supervised Encoding (L-CSSE) [72] which is a deep learning feature representation. The L-CSSE feature extractions involves a $l_{2,1}$ norm in auto-encoder based representation learning to promote joint sparsity among same-class samples. Majumdar et al. [72] have reported impressive face verification performance using L-CSSE features and SVM as classifier. In this experiment, HSRS-SVM is learned over 1,792 dimensional L-CSSE feature representations of face images with parameterization of $h = 5$ and $\mu = 25$.

Table 9 and Figure 14 provide accuracy comparison of LibSVM and HSRS-SVM with same L-CSSE feature representations. Further, accuracy values of some of the state-of-the-art approaches are also provided. To further analyze the classification performance difference between LibSVM

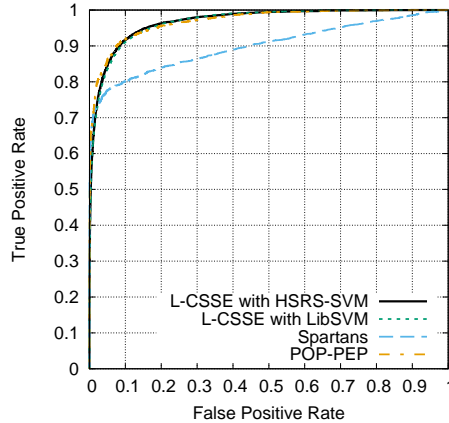


Figure 14: ROC curves on the restricted protocol of LFW dataset [19].

and HRSR-SVM, McNemar’s test ($p = 0.3613$) and paired t-test ($p = 0.0808$) between fold-wise
640 accuracy values are employed. Both the test indicate that, there is no statistically significant
difference between classification performance of LibSVM and HRSR-SVM. It is observed that the
proposed subclass reduced set based SVM required 2,972 seconds for training whereas, LibSVM
and Neural Network [72] required 3,288 and 3,382 seconds respectively on machine with Intel Xeon
Processor, 6 Core and 64GB RAM. The cardinality of estimated MRRS set is observed to be 4,732.
645 The cardinalities of $T_{r,SV}$ (support vectors at root level) and T_{SV} (support vectors of LibSVM)
are observed to be 1,809 and 1,874, respectively. It can be seen that the verification performance
of proposed HRSR-SVM with the deep learning based features is comparable to state-of-the-art
approaches. This provides an empirical evidence for the suitability of the proposed approach with
deep learning based features.

650 9. Conclusion and Future work

In this work we presented a novel approach for efficiently learning nonlinear support vector
machine classifier from large training data. The proposed approach obtains a set of candidate
support vectors based on computationally low-cost linear subproblems. We show that utilizing
these candidate support vectors (termed as estimated MRRS) to learn the overall nonlinear de-
655 cision boundary helps to reduce the overall training time significantly. Although, the proposed
approach relies on an approximation stage for estimating MRRS, the decision boundary and
classification accuracy are not significantly different than that of LibSVM. A hierarchical ex-
tension is also proposed, that divides the MRRS estimation task further into multiple iterative
stages. Experimental results are shown on several synthetic and real-world datasets including

660 `adult`, `ijcnn1`, `covertype`, `cifar-10`, and `LSL-FD`. Synthetic datasets are leveraged to gain the understanding of individual stages of the proposed approach and to compare the obtained decision boundaries with a traditional solver. We observe that the proposed approach yields two to five fold speed-up compared to LibSVM and almost up to an order of magnitude compared to other SVM-based large scale learning approaches. We also showcase the suitability of proposed
 665 HSRS-SVM approach with deep learning based features for face verification on `LFW` dataset.

Appendix A. Additional Toy Example

Here, we provide experiments performed on an additional toy dataset. The dataset is created by specifying the number of actual subclasses. As shown in Figure A.15 three version are created with $h = 10, 20$, and 100 . As h increases, visually, the degree of non-linearity is also increasing.
 670 As shown in Table A.10, the proposed approach consistently trains faster than LibSVM, across various values of h and n . This suggests that proposed approach is suitable with the classification problems involving distributions with large number of actual subclasses.

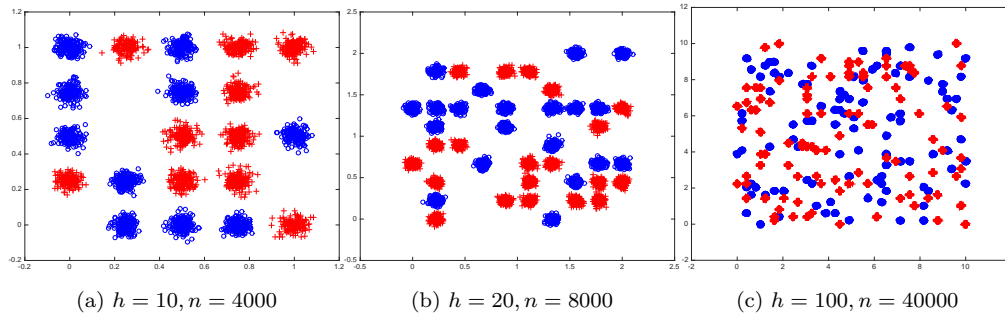


Figure A.15: (Best viewed in color) Synthetic datasets with varying number of subclasses (h) and samples (n). Each subclass consists of 200 samples.

Number of subclasses (h)	Number of samples (n)	Training Time of SRS-SVM (seconds)	Training Time of LibSVM (seconds)	Accuracy of SRS-SVM (%)	Accuracy of LibSVM (%)
10	4,000	28.54	37.84	100.00	100.00
20	8,000	98.74	150.31	100.00	100.00
100	40,000	693.73	2,799.31	99.97	100.00

Table A.10: Training time and classification accuracy of the synthetic datasets with varying number of subclasses, samples.

References

- [1] C. Cortes, V. Vapnik, Support-vector networks, *ML* 20 (3) (1995) 273–297.
- 675 [2] P. Gehler, S. Nowozin, On feature combination for multiclass object classification, in: *IEEE ICCV*, 221–228, 2009.
- [3] R. Singh, M. Vatsa, A. Noore, Integrated Multilevel Image Fusion and Match Score Fusion of Visible and Infrared Face Images for Robust Face Recognition, *Pattern Recogn.* 41 (3) (2008) 880–893.
- [4] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: *IEEE CVPR*, vol. 1, 680 886–893, 2005.
- [5] B. E. Boser, I. M. Guyon, V. N. Vapnik, A training algorithm for optimal margin classifiers, in: *CLT*, 144–152, 1992.
- [6] T. Joachims, Training linear SVMs in linear time, in: *SIGKDD*, 217–226, 2006.
- [7] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, S. Sundararajan, A dual coordinate descent 685 method for large-scale linear SVM, in: *ICML*, 408–415, 2008.
- [8] S. Shalev-Shwartz, Y. Singer, N. Srebro, A. Cotter, Pegasos: Primal estimated sub-gradient solver for SVM, *MP* 127 (1) (2011) 3–30.
- [9] G.-X. Yuan, C.-H. Ho, C.-J. Lin, Recent advances of large-scale linear classification, *Proceedings of the IEEE* 100 (9) (2012) 2584–2603.
- 690 [10] I. W. Tsang, J. T. Kwok, P.-M. Cheung, Core vector machines: Fast SVM training on very large data sets, in: *JMLR*, 363–392, 2005.
- [11] M. Zhu, A. M. Martinez, Subclass discriminant analysis, *IEEE TPAMI* 28 (8) (2006) 1274–1286.
- [12] N. Gkalelis, V. Mezaris, I. Kompatsiaris, T. Stathaki, Linear subclass support vector machines, *IEEE SPL* 19 (9) (2012) 575–578.
- 695 [13] S. Das, S. Datta, B. B. Chaudhuri, Handling data irregularities in classification: Foundations, trends, and future challenges, *Pattern Recognition* 81 (2018) 674–693.
- [14] J. C. Platt, Fast training of support vector machines using sequential minimal optimization, *Advances in kernel methods* (1999) 185–208.
- [15] D. Prokhorov, *IJCNN 2001 neural network competition*, Slide presentation in *IJCNN* 1 (2001) 97.
- 700 [16] A. Krizhevsky, *Learning Multiple Layers of Features from Tiny Images*, Tech. Rep., University of Toronto, 2009.
- [17] J. A. Blackard, D. J. Dean, Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables, *COMPAG* 24 (3) (1999) 131–151.
- 705 [18] S. Sonnenburg, V. Franc, E. Yom-Tov, M. Sebag, *Pascal large scale learning challenge*, URL <http://largescale.ml.tu-berlin.de> .
- [19] G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, *Labeled faces in the wild: A database for studying face recognition in unconstrained environments*, Tech. Rep., 07-49, University of Mas-

- sachusetts, Amherst, 2007.
- 710 [20] H. Yu, J. Yang, J. Han, Classifying large data sets using SVMs with hierarchical clusters, in: ACM SIGKDD, 306–315, 2003.
- [21] D. Boley, D. Cao, Training Support Vector Machines Using Adaptive Clustering., in: SDM, SIAM, 126–137, 2004.
- [22] H. P. Graf, E. Cosatto, L. Bottou, I. Dourdanovic, V. Vapnik, Parallel support vector machines: The cascade SVM, in: NIPS, 521–528, 2004.
- 715 [23] Y.-J. Lee, O. L. Mangasarian, RSVM: Reduced Support Vector Machines., in: SDM, vol. 1, 325–361, 2001.
- [24] K.-M. Lin, C.-J. Lin, A study on reduced support vector machines, IEEE TNN 14 (6) (2003) 1449–1459.
- 720 [25] P. Ilayaraja, N. Neeba, C. Jawahar, Efficient implementation of SVM for large class problems, in: ICPR, 1–4, 2008.
- [26] J. Wang, P. Neskovic, L. N. Cooper, Training data selection for support vector machines, in: Advances in natural computation, Springer, 554–564, 2005.
- [27] J. S. Nath, S. K. Shevade, An efficient clustering scheme using support vector methods, Pattern Recognition 39 (8) (2006) 1473–1480.
- 725 [28] C.-J. Hsieh, S. Si, I. Dhillon, A Divide-and-Conquer Solver for Kernel Support Vector Machines, in: ICML, 566–574, 2014.
- [29] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, JMLR 2 (2002) 45–66.
- 730 [30] N. Syed, H. Liu, K. Sung, Incremental learning with support vector machines, in: IJCAI, 1999.
- [31] K. Lau, Q. Wu, Online training of support vector classifier, Pattern Recognition 36 (8) (2003) 1913–1920.
- [32] L. Ralaivola, F. dAlché Buc, Incremental support vector machine learning: A local approach, in: ICANN, 322–330, 2001.
- 735 [33] T. Poggio, G. Cauwenberghs, Incremental and decremental support vector machine learning, NIPS 13 (2001) 409.
- [34] M. Karasuyama, I. Takeuchi, Multiple incremental decremental learning of support vector machines., IEEE TNN 21 (7) (2010) 1048–1059.
- [35] H. Mehrotra, R. Singh, M. Vatsa, B. Majhi, Incremental granular relevance vector machine: A case study in multimodal biometrics, Pattern Recognition 56 (2016) 63–76.
- 740 [36] J. A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, NPL 9 (3) (1999) 293–300.
- [37] L. Bottou, C.-J. Lin, Support vector machine solvers, Large scale kernel machines (2007) 301–320.
- [38] J. Langford, L. Li, A. Strehl, Vowpal wabbit online learning project, 2007.

- 745 [39] X. Huang, S. Mehrkanoon, J. A. Suykens, Support vector machines with piecewise linear feature mapping, *Neurocomputing* 117 (2013) 118–127.
- [40] M. Fornoni, B. Caputo, F. Orabona, Multiclass latent locally linear support vector machines, in: *ACML*, 229–244, 2013.
- [41] L. Ladicky, P. Torr, Locally linear support vector machines, in: *ICML*, 985–992, 2011.
- 750 [42] V. Kecman, J. P. Brooks, Locally linear support vector machines and other local models, in: *IEEE IJCNN*, 1–6, 2010.
- [43] T. B. Johnson, C. Guestrin, Unified Methods for Exploiting Piecewise Linear Structure in Convex Optimization, in: *NIPS*, 4754–4762, 2016.
- [44] E. Y. Chang, K. Zhu, H. Wang, H. Bai, J. Li, Z. Qiu, H. Cui, PSVM: Parallelizing Support Vector
755 Machines on Distributed Computers, in: *NIPS*, 2007.
- [45] S. Tyree, J. R. Gardner, K. Q. Weinberger, K. Agrawal, J. Tran, Parallel Support Vector Machines in Practice, arXiv preprint arXiv:1404.1066 .
- [46] L. Zanni, T. Serafini, G. Zanghirati, Parallel software for training large scale support vector machines on multiprocessor systems, *JMLR* 7 (2006) 1467–1492.
- 760 [47] T.-N. Do, F. Poulet, Classifying one billion data with a new distributed SVM algorithm., in: *RIVF*, 59–66, 2006.
- [48] C. Caragea, D. Caragea, V. Honavar, Learning support vector machines from distributed data sources, in: *AAAI*, 4, 1602, 2005.
- [49] P. A. Forero, A. Cano, G. B. Giannakis, Consensus-based distributed support vector machines,
765 *JMLR* 11 (2010) 1663–1707.
- [50] T.-N. Do, F. Poulet, Parallel Learning of Local SVM Algorithms for Classifying Large Datasets, *TLDKS* (2017) 67–93.
- [51] W. Guo, N. K. Alham, Y. Liu, M. Li, M. Qi, A Resource Aware MapReduce Based Parallel SVM for Large Scale Image Classifications, *NPL* 44 (1) (2016) 161–184.
- 770 [52] V. Sindhwani, S. S. Keerthi, Large scale semi-supervised linear SVMs, in: *ACM SIGIR*, 477–484, 2006.
- [53] S. S. Keerthi, O. Chapelle, D. DeCoste, Building support vector machines with reduced classifier complexity, *JMLR* 7 (2006) 1493–1515.
- [54] Q. Le, T. Sarlós, A. Smola, Fastfood-approximating kernel expansions in loglinear time, in: *ICML*,
775 2013.
- [55] T. Hastie, R. Tibshirani, Discriminant analysis by Gaussian mixtures, *JSTOR B* (1996) 155–176.
- [56] D. Reynolds, Gaussian mixture models, *Encyclopedia of Biometrics* (2015) 827–832.
- [57] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise., in: *KDD*, vol. 96, 226–231, 1996.
- 780 [58] R. Sibson, SLINK: an optimally efficient algorithm for the single-link cluster method, *The computer*

- journal 16 (1) (1973) 30–34.
- [59] D. Defays, An efficient algorithm for a complete link method, *The Computer Journal* 20 (4) (1977) 364–366.
- [60] S. Lloyd, Least squares quantization in PCM, *IEEE Transactions on Information Theory* 28 (2) (1982) 129–137.
- 785 [61] I. Rodriguez-Lujan, C. Santa Cruz, R. Huerta, Hierarchical linear support vector machine, *Pattern Recognition* 45 (12) (2012) 4414–4427.
- [62] D. Wang, X. Zhang, M. Fan, X. Ye, Hierarchical mixing linear support vector machines for nonlinear classification, *Pattern Recognition* 59 (2016) 255–267.
- 790 [63] M. E. Mavroforakis, S. Theodoridis, A geometric approach to support vector machine (SVM) classification, *IEEE TNN* 17 (3) (2006) 671–682.
- [64] M. Zhu, A. M. Martínez, Optimal subclass discovery for discriminant analysis, in: *IEEE CVPRW*, 97–97, 2004.
- [65] M. Á. Carreira-Perpiñán, C. K. Williams, On the number of modes of a Gaussian mixture, in: *SSTCV*, 625–640, 2003.
- 795 [66] M. A. T. Figueiredo, A. K. Jain, Unsupervised learning of finite mixture models, *IEEE TPAMI* 24 (3) (2002) 381–396.
- [67] N. Ueda, R. Nakano, Z. Ghahramani, G. E. Hinton, SMEM algorithm for mixture models, in: *NIPS*, 599–605, 1999.
- 800 [68] R. Collobert, S. Bengio, Y. Bengio, A parallel mixture of SVMs for very large scale problems, *Neural computation* 14 (5) (2002) 1105–1114.
- [69] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, *ACM TIST* 2 (3) (2011) 27.
- [70] K. Zhang, L. Lan, Z. Wang, F. Moerchen, Scaling up kernel SVM on limited resources: A low-rank linearization approach, in: *AISTat*, 1425–1434, 2012.
- 805 [71] N. Djuric, L. Lan, S. Vucetic, Z. Wang, BudgetedSVM: A toolbox for scalable SVM approximations, *JMLR* 14 (1) (2013) 3813–3817.
- [72] A. Majumdar, R. Singh, M. Vatsa, Face Recognition via Class Sparsity based Supervised Encoding, *IEEE TPAMI* 39 (6) (2017) 1273 – 1280.
- 810 [73] F. Juefei-Xu, K. Luu, M. Savvides, Spartans: single-sample periocular-based alignment-robust recognition technique applied to non-frontal scenarios, *IEEE TIP* 24 (12) (2015) 4780–4795.
- [74] H. Li, G. Hua, Hierarchical-PEP model for real-world face recognition, in: *IEEE CVPR*, 4055–4064, 2015.
- [75] S. R. Arashloo, J. Kittler, Class-specific kernel fusion of multiple descriptors for face verification using multiscale binarised statistical image features, *IEEE TIFS* 9 (12) (2014) 2100–2109.
- 815