# ENSEMBLE RANDOM-SUBSET SVM

Anonymous for Review

Keywords: Ensemble Learning, Bagging, Boosting, Generalization Performance, Support Vector Machine

Abstract: In this paper, the Ensemble Random-Subset SVM algorithm is proposed. In a random-subset SVM, multiple SVMs are used, and each SVM is considered a weak classifier; a subset of training samples is randomly selected for each weak classifier with randomly set parameters, and the SVMs with optimal weights are combined for classification. A linear SVM is adopted to determine the optimal kernel weights; therefore, an ensemble random-subset SVM is based on a hierarchical SVM model. An ensemble random-subset SVM outperforms a single SVM even when using a small number of samples (10 or 100 samples out of 20,000 training samples for each weak classifier); in contrast, a single SVM requires more than 4,000 support vectors.

## 1 INTRODUCTION

Although support vector machines (SVMs)(Vapnik, 1998)(Schölkopf et al., 1999)(Cristiani and Taylor, 2000) provide high accuracy and generalization performance, large amount of computation time and memory are required when they are applied to large-scale problems. Therefore, many previous works attempted to resolve these problems. For example, Chapelle(Chapelle, 2007) examined the effect of optimization of SVM algorithm, which reduces the computation complexity from $O(n^3)$ (for naive implementation) to about $O(n_{sv}^2)$ ($n_{sv}$ denotes the number of support vectors), and Keerthi(Keerthi at al., 2007) proposed to reduce the number of kernels with forward stepwise selection and attained a computational complexity of $O(nd^2)$, where $d$ denotes the number of selected kernels. Lin(Lin and Lin, 2003) proposed to select a random subset of the training set; however, this approach could not reduce the number of basis functions to attain an accuracy close to that of a full SVM solution. Demir(Demir, 2007) applied the RANSAC(Fischler et al.,1981) algorithm to reduce the number of training samples for the Relevance-Vector Machine (RVM) for remote sensing data. Nishida proposed the RANSAC-SVM(Nishida and Kurita, 2008) to reduce the number of training samples for improving generalization performance. In the RANSAC-SVM, an SVM is trained using a randomly selected subset of training samples, and hyperparameters (such as a regularization term and Gaussian width) are set to fit over whole training samples. Though RANSAC-SVM effectively reduced the computation time required for training by reducing the number of training samples, a large number of trials were required to determine a combination of *good* subset samples and *good* hyperparameters (e.g., 100,000 trials were performed in (Nishida and Kurita, 2008)). The hyperparameters for a single SVM must be strictly determined to simultaneously achieve high classification accuracy and good generalization performance. Therefore, RANSAC-SVM had to perform an exhaustive search for *good* parameters.

Many ensemble learning algorithms have been proposed for improving the generalization performance such as boosting(Freund and Schapire, 1996) and bagging(Breiman, 1998). In ensemble learning, the weak classifiers must maintain high generalization performance, but high classification accuracy is not required for each weak classifier because an accurate classification boundary is determined by combining the weak (low classification accuracy) classifiers. Therefore, we have an opportunity of avoiding an exhaustive search for the appropriate hyperparameters,

by providing a variety of weak classifiers. In this paper, we propose the *Ensemble Random-Subset SVM*, which is a natural extension of the RANSAC-SVM, for ensemble learning.

The rest of the paper is organized as follows. In Section 2, we describe the Ensemble Random-Subset SVM algorithm. In Section 3, we present the experimental results for an artificial dataset. Ensemble Random-Subset SVM showed good classification performance and outperformed a single SVM for the same test samples.

# 2 ENSEMBLE RANDOM-SUBSET SVM

The algorithm of an ensemble random-subset SVM is presented in this section. We first introduce the previous works on learning algorithms that use subsets of training samples, and then, we introduce the definition of an ensemble kernel SVM, that uses subsets of training samples.

## 2.1 Learning Algorithms Using Subsets of Training Samples

Several algorithms that use a subset of training samples have been proposed previously. These algorithms can be used to improve the generalization performance of classifiers or to reduce the computation cost for the training. *Feature vector selection* (FVS) (Baudat, 2003) has been used to approximate the feature space $F$ spanned by training samples by the subspace $F_s$ spanned by selected *feature vectors* ($FVs$). The *import vector machine* (IVM) is built on the basis of kernel logistic regression and is used to approximate the kernel feature space by a smaller number of *import vectors* (IVs). Whereas FVS and IVM involve the approximation of the feature space by their selected samples, *RANSAC-SVM* (Nishida and Kurita, 2008) involves the approximation of the classification boundary by randomly selected samples with optimal hyperparameters. In the cases of FVS and IVM, the samples are selected sequentially, but in the case of RANSAC-SVM, samples are selected randomly; nevertheless, in all these cases, a single kernel function is used over all the samples.

SABI (Oosugi and Uehara, 1998) sequentially selected a pair of samples at a time and carried out linear interpolation between the samples in the pair in order to determine a classification boundary. Although SABI does not use the kernel method, the combination of classification boundaries in SABI can be considered as a combination of different kernels.

An exhaustive search for the optimal sample subset requires large amount of computation time; therefore, we employed random sampling to select subsets and combined multiple kernels with different hyperparameters for the subsets for ensemble random-subset SVM.

## 2.2 Ensemble Kernel SVM

The classification function is given as

$$f(x) = \mathrm{sign}(w^T \phi(x) - h). \tag{1}$$

where function $\mathrm{sign}(u)$ is a sign function, which outputs 1 when $u > 0$ and outputs -1 when $u \leq 0$; $w$ denotes a weight vector of the input; and $h$ denotes a threshold. $\phi(x)$ denotes a nonlinear projection of an input vector, such as $\phi(x_1)^T \phi(x_2) = K(x_1, x_2)$. $K$ is called a *Kernel Function* and is usually a simple function, such as the Gaussian function

$$K(x_1, x_2) = \exp\left(\frac{-||x_1 - x_2||^2}{G}\right). \tag{2}$$

Substituting equation (2) in equation (1), we obtain the following classification function:

$$f(x) = \mathrm{sign}(\sum_i \alpha_i t_i K(x_i, x) - h, \tag{3}$$

where $\alpha_i$ denotes the sample coefficients.

On the basis of equation (3), the classification function for ensemble kernel SVM is determined as follows:

$$f(x) = \sum_{m=1}^{p} \beta_m \sum_{i=1}^{n} \alpha_{m_i} t_i K(x_i, x) - h, \tag{4}$$

where $n$ is the number of sample; $\alpha_{m_i}$, the weight coefficient; and $t_i$, the sample label. The kernel weights satisfy the conditions $\beta_m \geq 0$ and $\sum_{m=1}^{p} \beta_i = 1$. Different kernels (such as linear, polynomial, and Gaussian kernels) or kernels with different hyper-parameters (for example, Gaussian kernels with different Gaussian widths) can be combined; however, the same weight is assigned to a kernel over all the input samples, as per the definition in equation (4).

In the ensemble SVM described in equation (4), all the training samples are used to determine a kernel matrix. However, kernels over different subsets of training samples can be combined; for example,

$$f(x) = \sum_{m=1}^{p} \beta_m \sum_{i \in \dot{X}} \alpha_{m_i} t_i \langle K_m(x, x_i) \rangle + h, \tag{5}$$

where $\dot{X}$ denotes the subset of training samples for the $m$th kernel, and $X$ denotes the full set of training

samples. The sampling policy for the subsets is not restricted to any method, but if the subsets are sampled according to the probability distribution $\eta_m(x)$, the kernel matrix is defined as follows:

$$K_\eta(\dot{x}_i, \dot{x}_j) = \sum_{m=1}^{p} \langle \Phi_m(\dot{x}_i), \Phi_m(\dot{x}_j) \rangle, \qquad (6)$$

where $\dot{X} = \eta X$. The probability that $K_\eta(\dot{x}_i, \dot{x}_j)$ is obtained becomes the product of the probabilities of obtaining $x_i$ and $x_j$.

## 2.3 Subset Sampling and Training Procedure

Because the subset kernel ($K_m$) is determined by the subset of training samples ($\dot{X}_m$), the subset selection strategy may affect the classification performance of each kernel. Therefore, in a random-subset SVM, the following three parameters must be optimized: sample weight $\alpha_{m_i}$, kernel weight $\beta_m$, and sample subset $\dot{X}_m$. However, since simultaneous optimization of three parameters is a very complicated process, we generate randomly selected subsets to determine $\alpha_{m_i}$s for a subset kernel with randomly assigned hyperparameters; then, we determine $\beta_m$ as the optimal weight for each kernel. When the kernel weights $\beta_m$ are maintained to be optimal, the weights for kernels with insufficient performance become low. Therefore, such kernels may not affect the overall performance.

A RBF SVM is employed for each weak classifier $f_m(x)$, and an ensemble random-subset SVM is implemented in the form of a hierarchical SVM. Therefore, we first optimize the sample weights $\alpha_i$ for each subset-kernel SVM $f_m(x)$ and then optimize the classifier weights $\beta_m$. We employed the additive approach for determining a new weak classifier to maintain the generalization performance for the integrated classifier. The detailed algorithm is as follows:

1. Let $n$ be the number of training samples $X$; $M$, be the number of kernels; $Q$, be the number of samples in the selected subsets $\dot{X}_m$; and $R$, be the number of trials for the parameter selection
2. Repeat the following steps $M$ times ($\{m = 1 \ldots M\}$)
   (a) Repeat the following steps $R$ times ($\{r = 1 \ldots R\}$)
      i. Determine a training subset $\dot{X}_m^r$ by randomly selecting $Q$ samples from $X$
      ii. Randomly set hyperparameters (such as the Gaussian width and the regularization term for the RBF kernel)
      iii. Train temporally the $m$th classifier $f_m^r$ over the subset $\dot{X}_m^r$
      iv. Predict all training samples $X$ using $f_m^r$, to determine the probability output
      v. Train a linear SVM over $\{f_1 \ldots f_{m-1}, f_m^r\}$ to determine the optimal $\beta_m^r$ in order to obtain the temporal integrated classifier $F_m^r$
   (b) Select the $f_m^r$ that gives the optimal $F_m^r$ to be the $m$th weak classifier $f_m$
3. Train a linear SVM over $f_1 \ldots f_M$ to determine the optimal $\beta_M$ in order to obtain the final classifier $F_M$

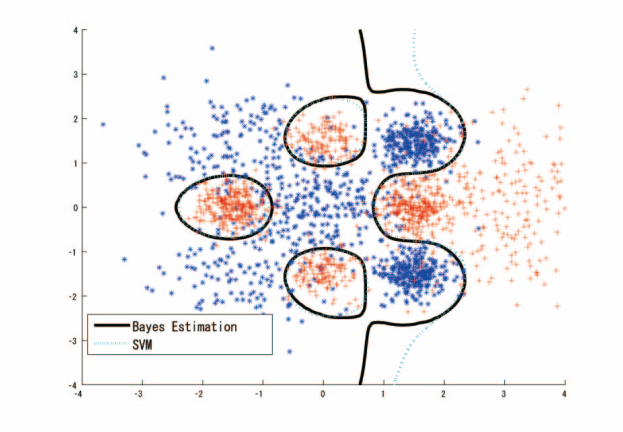


Figure 1: Experimental Data and Classification Boundary. *Black line indicates classification boundary for Bayesian estimation. Blue dashed line indicates classification boundary for a single SVM.*

# 3 EXPERIMENTAL RESULTS

The experimental results are discussed in this section. Although a wide variety of kernels are suited for use in ensemble random-subset SVM, we use only RBF-SVM for the subset SVMs to investigate the effect of random sampling. Hyperparameters ($G$ and $C$ for LIBSVM (Chang and Lin, 2001)) are randomly set to the desired range for the dataset. We employed linear SVM to combine the subset kernels in order to obtain the optimal kernel weight for classification.

## 3.1 Experimental Data

We evaluated the ensemble random-subset SVM by using the artificial data in this experiment. The data are generated from a mixture of ten Gaussian distributions; of these, five generate class 1 samples, and the other five generate class –1 samples. 10,000 samples are generated for each class as training data, and another 10,000 samples are independently generated for

each class as test data. The contour in figure 1 indicates the Bayesian estimation of the class boundary; the classification ratio for the Bayesian estimation is 92.25% for the training set and 92.15% for the test set. The classification ratio for the full SVM, in which the parameters are determined by five-fold cross validation ($c = 3$ and $g = 0.5$), is 92.22% for the training set and 91.95% for the test set (figure 1), with 4,257 support vectors.

The fitting performance of a random-subset SVM may be affected by the size of the subset; therefore, we evaluated a small subset (10-sample subset) and a larger subset (100-sample subset). All the experiments were run thrice, and the results were averaged.

## 3.2 10-Sample Subset SVM

We first examined the 10-sample subset SVM. Five sets of parameters ($C$ and $G$) were generated randomly and evaluated for a 10-sample subset. Then, the parameter set that yielded the best classification ratio for all training samples was selected for the subset SVM. We generated five sample subset candidates at a time and thus evaluated 25 subset/parameter sets in the selection procedure.

Figure 2 shows the classification ratio for the training samples, and figure 3 shows the classification ratio for the test samples. The classification ratio for the training samples converged quickly, exceeding the Bayesian estimation for 40 kernels and finally reached 92.26% for 200 kernels. Although this indicated slight over-learning for the training samples, the classification ratio for the test samples indicated fairly good classification performance (comparable with the result of the full SVM) and reached 91.98% for 200 kernels.

The classification boundary in figure 4 also indicates stable classification performance for the 10-sample subset.

### 3.2.1 100-Sample Subset SVM

Figures 5 and 6 show the classification ratio for the 100-sample subset SVM with parameter selection for the training samples and the test samples respectively. The result showed a trend similar to that observed for the 10-sample subset-SVM; slight over-learning was observed for the training samples (92.26% for 200 kernels), and the classification ratio was similar to the SVM result for the test samples (91.98% at 200 kernels). As figure 7 shows, the classification boundary obtained by the 100-sample subset SVM is very similar to that obtained by the Bayesian estimation.

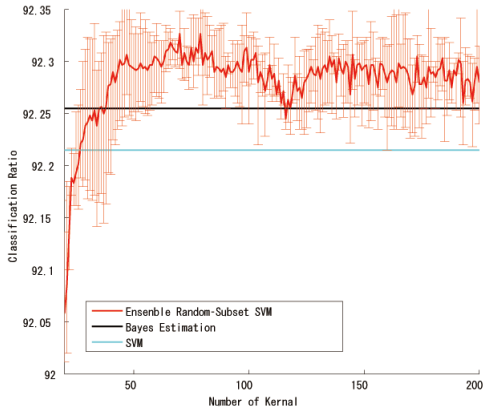


Figure 2: Result for 10-Sample Subset SVM (Training)

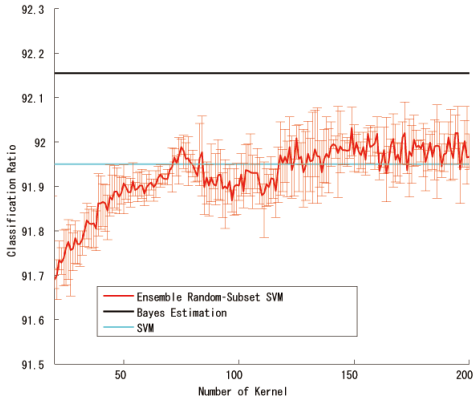


Figure 3: Result for 10-Sample Subset SVM (Test)

## 3.3 Result for Benchmark Set

Next, we examined a benchmark set `cod-rna` from the LIBSVM dataset (Libsvm-Dataset). The `cod-rna` dataset has eight attributes, 59,535 training samples, and 271,617 validation samples with two-class labels. Hyperparameters for a single SVM were obtained by performing a grid search through five-fold cross validation, whereas the hyperparameters for the ensemble random-subset SVM were set such that their values were close to the values for the single SVM. We applied the random-subset SVM for this dataset because the dataset includes a large number of samples. We examined 500-sample, 1000-sample, and 5000-sample subsets.

Table 1 shows the results for the `cod-rna` dataset. The ensemble random-subset SVM outperformed the single SVM with a subset size of 1,000 (1.7% of

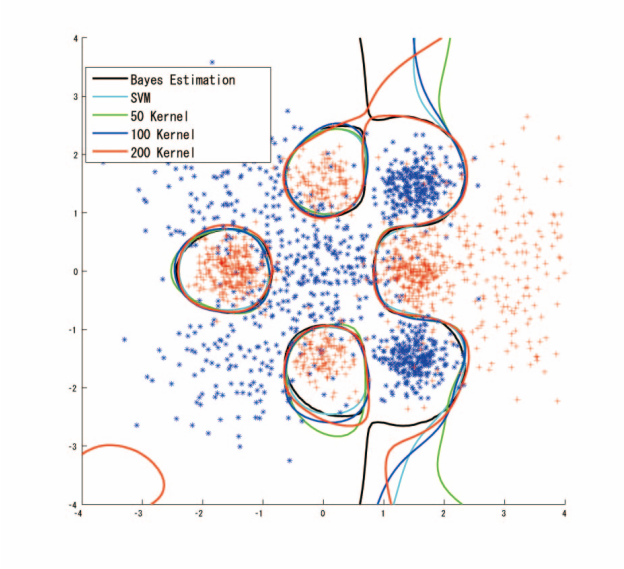


Figure 4: Classification Boundary for 10-Sample Subset SVM

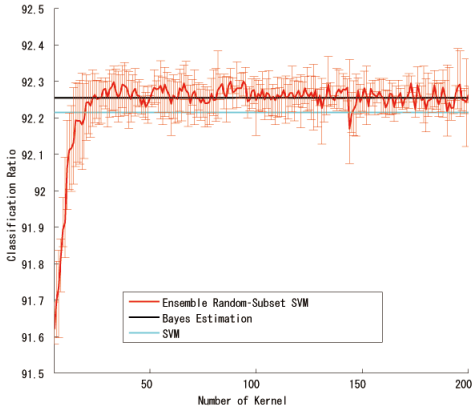


Figure 5: Result for 100-Sample Subset SVM (Training)

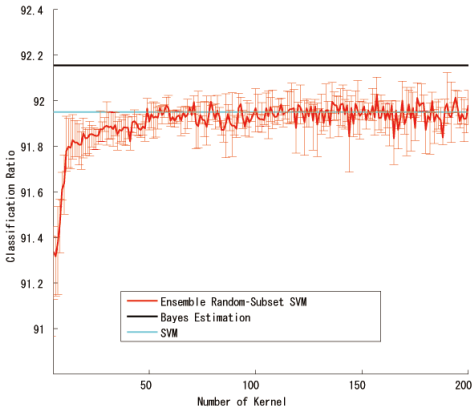


Figure 6: Result for 100-Sample Subset SVM (Test)

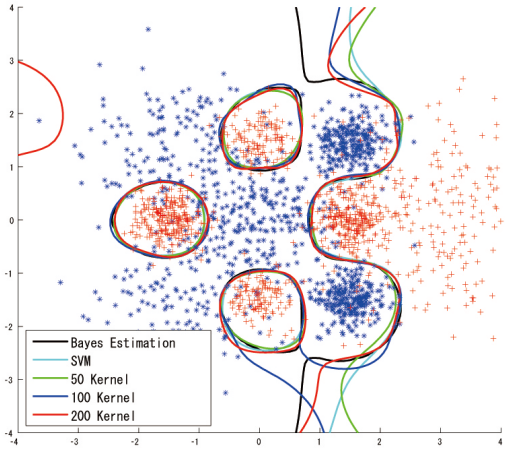


Figure 7: Classification Boundary for 100-Sample Subset SVM

the total number of the training samples) combining 2,000 SVMs and with a subset of 5,000 (8.3% of the training samples) combining 100 SVMs.

## 4 CONCLUSION

We proposed an ensemble random-subset SVM algorithm, which combines multiple kernel SVMs generated from small subsets of training samples.

The 10-sample subset-SVM outperformed the single SVM (4,257 support vectors), combining about 120 subset SVMs, and the 100-sample subset SVM also outperformed the single SVM, combining about 50 subset-SVMs. The use of a larger subset (100-sample subset) not only helped accelerate the convergence of the classifier but also slightly improved the final classification ratio.

The result for the benchmark dataset `cod-rna` showed that an ensemble random-subset SVM with a subset size of 2% or 5% of the training samples can outperform a single SVM with optimal hyperparameters.

Although 200 or 2000 SVMs must be combined in an ensemble random-subset SVM, the number of computations for the subset-kernels would not exceed

Table 1: Classification Ratio for `cod-rna` dataset

| | Number of kernels | Training | Test |
|---|---|---|---|
| Single SVM (Full set) | 1 | 95.12 | 96.23 |
| Ensemble SVM subset = 500 | 3000 | 95,03 | 96.16 |
| Ensemble SVM subset = 1000 | 2000 | 95.30 | **96.30** |
| Ensemble SVM subset = 5000 | 100 | 94.90 | **96.24** |

that for a single (full-set) SVM because an SVM requires at least $O(N^2)$ to $O(N^3)$ computations.

We employed a linear SVM to combine the kernels and obtain the optimal kernel weights. However, this final SVM took up a majority of the computation time of the ensemble random-subset SVM because it had to be trained for as many samples as the large-attribute training samples.

In this study, we used all the outputs from subset kernels for the training samples; however, we can apply feature selection and sample selection for the final linear SVM, as this may help reduce the computation time and improve the generalization performance simultaneously.

## REFERENCES


V.N.Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998.

B.Schölkopf, C.J.C.Burges, A.J.Smola, *Advances in Kernel Methods - Support Vector Learning*, The MIT Press, 1999.

N.Cristianini, J.S-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, 2000.

O.Chapelle, "Training a Support Vector Machine in the Primal", in *Large-Scale Kernel Machines*, pp.29-50, The MIT Press, 2007.

S.S.Keerthi, O.Chapelle, D.DeCoste, "Building SVMs with Reduced Classifier Complexity", in *Large-Scale Kernel Machines*, pp.251-274, The MIT Press, 2007.

K.-M.Lin, and C.-J.Lin, "A Study on Reduced Support Vector Machines", *IEEE Transactions on Neural Networks*, Vol.14, pp.1449-1459, 2003.

B. Demir, S. Erturk, "Hyperspectral Image Classification Using Relevance Vector Machines", *IEEE Geoscience and Remote Sensing Letters*, Vol.4, No.4, pp.586-590, 2007.

M.A.Fischler, R.C.Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Communications of the ACM*, Vol.24, pp.381-395, 1981.

Y.Freund, R.E.Schapire, "Experiments with a New Boosting Algorithm",in *Proc. of International Conf. on Machine Learning (ICML96)*, pp.148-156, 1996.

L.Breiman, "Bagging Predictors", Machine Learning, Vol.24, pp.123-140, 1996.

K.Nishida, T.Kurita, "RANSAC-SVM for Large-Scale Datasets", in *proc. International COnference on Pattern Recognition (ICPR2008)*, 2008. (CD-ROM).

G.Baudat, "Feature Vector Selection and Projection Using Kernels", in *NeuroComputing*, Vol.55, No.1, pp.21-38, 2003.

J.Zhu, T.Hastie, "Kernel Logistic Regression and the Import Vector Machine", *J.of Computational and Graphical Statistics*, Vol.14, No.1, pp.185-205, 2005.

Y.Oosugi, K.Uehara, "Constructing a Minimal Instance-base by Storing Prototype Instances", in *J. of Information Processing*, Vol.39, No.11, pp.2949-2959, 1998. (in Japanese).

C.C.Chang, C.J.Lin, "LIBSVM: a library for support vector machines", `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, 2001.

LIBSVM data set, `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#cod-rna`, 2006.