Multiple Random Subset-Kernel Learning

Kenji NISHIDA[1], Jun FUJIKI[1] and Takio KURITA[2]

[1] Human Technology Research Institute, National Institute of Advanced Industrial Science and Technology (AIST) Japan, 1-1-1 Umezono, Tsukuba Ibaraki, 305-8568, JAPAN,
{kenji.nishida,jun-fujiki}@aist.go.jp,
[2] Faculty of Engineering, Hiroshima University, Kagamiyama 1-7-1, Higashi-Hiroshima Hiroshima 739-8521, JAPAN,
tkurita@hiroshima-u.ac.jp

Abstract. In this paper, the multiple random subset-kernel learning (MRSKL) algorithm is proposed. In MRSKL, a subset of training samples is randomly selected for each kernel with randomly set parameters, and the kernels with optimal weights are combined for classification. A linear support vector machine (SVM) is adopted to determine the optimal kernel weights; therefore, MRSKL is based on a hierarchical SVM. MRSKL outperforms a single SVM even when using a small number of samples (200 to 400 out of 20,000 training samples), while the SVM requires more than 4,000 support vectors.

Keywords: Kernel Method, Multiple Kernel Learning, Support Vector Machine, Random Sampling

1 Introduction

Recently, multiple kernel learning (MKL) has been proposed to improve the classification performance of single kernel classifiers [1, 2]. Although the method based on the unweighted sum if multiple kernels is considered the simplest method, it may not be the ideal one.Therefore, various programming methods for finding the optimal combination weight have been proposed. Lanckreit [3] and Bach [1] proposed an efficient algorithm based on sequential minimal optimization (SMO).

The discriminant function for MKL is described as a weighted summation of kernel values:

$$f(x) = \sum_{m=1}^{p} \beta_m \langle \boldsymbol{w}_m, \boldsymbol{\Phi}(\boldsymbol{x}) \rangle + b \tag{1}$$

where m indexes kernels. β_m is the weight coefficients for the kernel; $\boldsymbol{w}_m$, the weight coefficient for the sample; $\boldsymbol{\Phi}_m(\boldsymbol{x})$, the mapping function for feature space m; and p, the number of kernels. Reforming equation (1) using the duality condition, we obtain

$$f(x) = \sum_{m=1}^{p} \beta_m \sum_{i=1}^{n} \alpha_{mi} y_i \underbrace{\langle \boldsymbol{\Phi}_m(\boldsymbol{x}), \boldsymbol{\Phi}_m(\boldsymbol{x}_i) \rangle}_{\boldsymbol{K}_m(\boldsymbol{x}, \boldsymbol{x}_i)} + b \tag{2}$$

where n is the number of sample; α_{mi}, the weight coefficient; and y_i, be the sample label. The kernel weights satisfy the condition $\beta_m \geq 0$ and $\sum_{m=1}^{p} \beta_i = 1$. Different kernels (such as linear, polynomial, and Gaussian kernels) or kernels with different hyperparameters (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 (1).

Although in the original definition of MKL (equation (1)) different weights are not assigned to a kernel for different samples, kernels can be combined over different subsets of training samples, such as

$$f(x) = \sum_{m=1}^{p} \beta_m \sum_{i \in \dot{\boldsymbol{X}}} \alpha_{mi} y_i \langle \boldsymbol{K}_m(\boldsymbol{x}, \boldsymbol{x}_i) \rangle + b \tag{3}$$

where $\dot{\boldsymbol{X}}$ stands for the subset of training samples for the mth kernel, while $\boldsymbol{X}$ stands for the full set of training samples. The sampling policy for the subsets is not restricted to any method, but if subsets are sampled according to the probability distribution $\eta_m(\boldsymbol{x})$, the kernel matrix is defined as follows:

$$K_\eta(\dot{\boldsymbol{x}}_i, \dot{\boldsymbol{x}}_j) = \sum_{m-1}^{p} \langle \Phi_m(\dot{\boldsymbol{x}}_i), \Phi_m(\dot{\boldsymbol{x}}_j) \rangle \tag{4}$$

where $\dot{\boldsymbol{X}} = \eta \boldsymbol{X}$. The probability that $K_\eta(\dot{\boldsymbol{x}}_i, \dot{\boldsymbol{x}}_j)$ is obtained becomes the product of the probabilities of obtaining $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Therefore, a subset kernel is determined by using the kernel matrix for all training samples and the sampling function η_m, as follows:

$$\begin{aligned} K_\eta(\dot{\boldsymbol{x}}_i, \dot{\boldsymbol{x}}_j) &= \sum_{m-1}^{p} \langle \Phi_m(\dot{\boldsymbol{x}}_i), \Phi_m(\dot{\boldsymbol{x}}_j) \rangle \\ &= \sum_{m=1}^{p} \eta_m(\boldsymbol{x}_i) \underbrace{\langle \Phi_m(\boldsymbol{x}_i), \Phi_m(\boldsymbol{x}_j) \rangle}_{\boldsymbol{K}_m(\boldsymbol{x}_i, \boldsymbol{x}_j)} \eta_m(\boldsymbol{x}_j), \end{aligned} \tag{5}$$

which is eventually equivalent to the definition of *localized multiple kernel learning* (LKML) [5].

For good classification performance in MKL, the optimal hyperparameters for the kernels and sample subsets (sampling function according to $\eta_m(\boldsymbol{x})$) must be determined by using different subsets; however, this requires an exhaustive search for the desired parameters and sampling functions. Therefore, we employ random sampling for the training subsets and randomly set hyperparameters for the kernels. The final classifier is determined by a linear combination of random kernels (randomly sampled subset and randomly set hyperparameters), and the β_m values are optimized to obtain the best classification performance.

In this paper, we propose *multiple random subset kernel learning* (MRSKL), a multiple kernel learning algorithm for a randomly selected subset of training samples. The

proposed algorithm uses a small subset for each kernel, and the kernel values are combined according to the classification result obtained for all training samples. Simultaneous optimization of α_{mi} and β_m in equation (3) has been a major interest in MKL research, as reported by Bach [1] and Rakotmamonjy [4], but the coefficients are independently optimized in the proposed algorithm.

The rest of the paper is organized as follows.

In Section 2, we describe the MRSKL algorithm. In Section 3, we present the experimental results for an artificial dataset. MRSKL showed good classification performance which exceeds the SVM result for the test samples.

2 Multiple Random Subset-Kernel Learning Algorithm

2.1 Learning Algorithms Using a Subset of Training Samples

Several algorithms that use a subset of training samples are proposed. 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) [6] has been used to approximate the feature space F spanned by training samples by the subspace F_s spanned by selected *feature vectors* (FVs). *Import vector machine* (IVM) is built on the basis pf kernel logistic regression (KLR) and used to approximate kernel feature space by a smaller number of *import vectors* (IVs). While FVS and IVM involve approximation of the feature space by their selected samples, *RANSAC-SVM* [9] involves approximation of the classification boundary by randomly selected samples with optimal hyperparameters. In FVS and IVM, samples are selected sequentially, but in the case of RANSAC-SVM, samples are randomly selected; nevertheless, in all these cases, a single kernel function is used over all the samples.

SABI [8] sequentially selected a pair of samples at a time and carried out linear interpolation between the pair in order to determine a classification boundary. Although SABI does not use the kernel method, the combination of classification boundaries can be considered as a combination of different kernels.

An exhaustive search for the optimal sample subset requires a large computation; therefore, we employed random sampling to select subsets and combined multiple kernels with different hyperparameters for the subsets for MRSKL.

2.2 Subset Sampling and Training Procedure for MRSKL

Since the subset-kernel ($\boldsymbol{K}_m$) is determined by the subset of training samples (S_m), the subset selection strategy may affect the classification performance of each kernel. Therefore, in MKL using subset-kernels, the following three parameters must be optimized; sample weight α_{mi}, kernel weight β_m, and sample subset S_m. However, since simultaneous optimization of three parameters is a very complicated process, we generate randomly selected subsets to determine α_{mi}s for a subset kernel with randomly assigned hyperparameters; then, we determine β_m as the optimal weight for each kernel. When the kernel weights β_m are maintained to be optimal, the weights for kernels with insufficient performance becomes low. Therefore, such kernels may not affect the overall performance.

Separating the optimization procedures for α_i (sample weight) and β_m (kernel weight), we rewrite equation (2) by substituting $\alpha_i y_i \langle \boldsymbol{K}_m(\boldsymbol{x}, \boldsymbol{x}_i) \rangle$ with $f_m(x)$, as follows:

$$\begin{aligned} f(x) &= \sum_{m=1}^{p} \beta_m \sum_{i \in S_m} \alpha_i y_i \langle \boldsymbol{K}_m(\boldsymbol{x}, \boldsymbol{x}_i) \rangle + b \\ &= \sum_{m=1}^{p} \beta_m f_m(x) + b \end{aligned} \tag{6}$$

In MRSKL, we first optimize α_i for the subset-kernel classifier $f_m(x)$ and then optimize β_m.

The detailed MRSKL algorithm is as follows:

1. Let n be the number of training samples T; p, the number of kernels; and l, the number of samples in the selected subsets S_m,
2. Repeat the following steps p times
   (a) Determine Q training subsets S_m by randomly selecting samples from T
   (b) Randomly set hyperparameters (such as Gaussian width and regularization term for the RBF kernel)
   (c) Train the mth classifier f_m over the subset S_m
   (d) Predict all training samples T by f_m determining probability output
3. Train a linear SVM over f_m: $\{m = 1 \ldots P\}$ to determine the optimal β_m for the final classifier

Parameter selection is performed by repeating steps 2b to step 2d, and the best parameter set is adopted in step 3.

RBF-SVM is employed for $f_m(x)$, and MRSKL is performed on the baisi of a hierarchical SVM.

3 Experiment

The experimental results are discussed in this section. Although a wide variety of kernels are suited for use in MRSKL, we use only RBF-SVM for the subset-kernels to investigate the effect of random sampling. Hyperparameters (G and C for LIBSVM [10]) are randomly set to the desired range for the dataset. We employed linear-SVM to combine subset kernels to obtain the optimal kernel weight for classification.

3.1 Experimental Data

We evaluated MRSKL by using the artificial data in this experiment. The data are generated from a mixture of ten Gaussian distributions, five of which generate class 1 samples and others generate class –1 samples. 20,000 samples are generated for the training set, and 20,000 samples are independently generated for test set. The black contour in the figure 1 indicates Bayesian estimation of the class boundary; the classification ratio for

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 and 91.95% for the test set, with 4,257 support vectors.

The fitting performance of MRSKL may be affected by the subset selection policy; therefore, we first evaluated the performance by the smallest subset size, which includes one pair of samples from class 1 and class –1 each. All the experiments were run thrice, and the results were averaged.

3.2 A Single-Pair Subset-Kernel

Figure 1 shows the classification boundary in MRSKL for various numbers of kernels. From the result, a good classification boundary can be determined using as few as 100 samples (50 single-pair subset-kernels), while a larger number of samples would be required in an SVM.

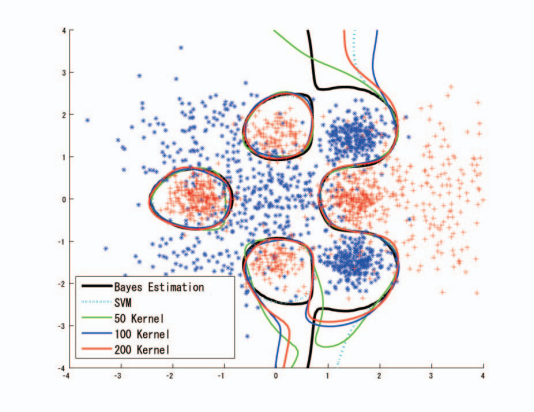

Fig. 1. MRSKL Classification Boundary with Single-Pair Kernel ($C = 2^{10}$ to 2^{-1}, $G = 2^2$ to 2^{-5})

Figure 2 shows the classification ratio for the training samples, and figure 3 shows the classification ratio for the test samples with the regularization parameter C for 2^{10} to 2^{-1} and the Gaussian width parameter G for 2^2 to 2^{-5}. The average classification ratio for the training samples became comparable to the SVM result for about 100 kernels but the classification ratio for the test samples exceeds the SVM result for 150 kernels. finally, the classification ratio reached 92.20% for 200 kernels. The average classification ratio for the test samples exceeded the SVM result for about 150 kernels and finally reached 91.97% for 200 kernels. Since each subset contained only one pair (two) of samples in this experiment, only 200 samples were required to attain a fitting performance similar ti that in the SVM case with 4,257 support vectors. This result for

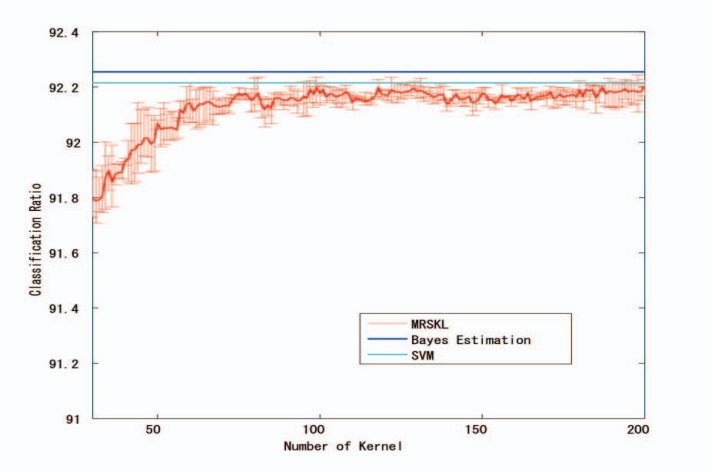

Fig. 2. MRSKL Result for Training Samples with Single-Pair Kernel ($C = 2^{10}$ to 2^{-1}, $G = 2^2$ to 2^{-5})

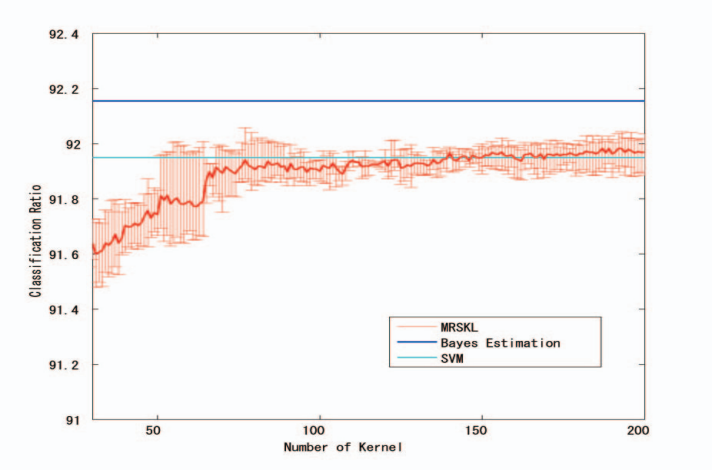

Fig. 3. MRSKL Result for Test Samples with Single-Pair Kernel ($C = 2^{10}$ to $2^{-}1$, $G = 2^2$ to 2^{-5})

the training samples indicates that MRSKL can show high fitting performance with a small number of support vectors than does an SVM. The result for the test samples indicates that MRSKL can show higher generalization performance than does an SVM.

3.3 Result for Benchmark Set

Next, we examined a benchmark set `cod-rna` from the LIBSVM dataset [11]. 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 grid search through five-fold cross-validation and randomly set for MRSKL around the values for a single SVM. We applied the random subset-kernel with parameter selection 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. MRSKL outperformed the single SVM with a subset size of 1,000 (1.7% of the total number of the training samples) combining 2,000 kernels and with a subset of 5,000 (8.3% of the training samples) combining 100 kernels.

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

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

4 Conclusion

We proposed an MRSKL algorithm, which combines multiple kernels generated from small subsets of training samples.

The result for the smallest subset (one pair) showed that MRSKL could approximate the classification boundary with a small number of samples. The 200-pair (400-samples) subset-kernel outperformed the SVM with 4,257 support vectors.

The result for the benchmark dataset `cod-rna` showed that MRSKL with a subset size corresponding to 2% or 5% of the training samples can outperform the single SVM with optimal hyper-parameters.

Although in MRSKL, 200 or 1000 kernels must be combined, the number of computations for the subset-kernels would not exceed 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 kernels and obtain the optimal kernel weights. However, this final SVM took up a majority of the computational time in

MRSKL since it had to 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 computation and improve the generalization performance simultaneously.

Acknowledgment

The authors would like to thank Prof. Toshikazu Wada of Wakayama University for providing us with valuable information on the relationship between prototyping and classification and also for drawing our attention to interesting papers.

This work was supported by JSPS KAKENHI 22500172.

References

1. F.R. Bach, G.R.G. Lanckriet, M.I. Jordan, "Multiple Kernel Learning, Conic Duality, and the SMO algorithm", in *Proc. International Conf. on Machine Learning*, pp.41-48, 2004.
2. S. Sonnnenburg, G. Rötsch, C. Schäfer, B. Schölkopf, "Large Scale Multiple Kernel Learning", in *J.of Machine Learning Research*, Vol.7, pp.1531-1565, 2006.
3. G.R.G. Lanckriet, N. Cristiani, P. Bartlett, L.E. Ghaoui, M.I. Jordan, "Learning the kernel matrix with semidefinite programming", in *J. of Machine Learning Research*, Vol.5, pp.27-72, 2004.
4. A. Rakotomamonjy, F. Bach, S. Canu, Y. Grandvalet, "More Efficiency in Multiple Kernel Learning", in *Proc. of International Conf. on Machine Learning*, 2007.
5. M. Gönen, E. Alpaydin, "Localized Multiple Kernel Learning", in *Proc. if International Conf. on Machine Kearning*, 2008.
6. G.Baudat, "Feature vector selection and projection using kernels", in *NeuroComputing*, Val.55, No.1, pp.21-38, 2003.
7. J.Zhu, T.Hastie, "Kernel Logistic Regression and the Import Vector Machine", in *J.of Computational and Graphical Statistics*, Vol.14, No.1, pp.185-205, 2005.
8. 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).
9. K.Nishida, T.Kurita, "RANSAC-SVM for Large-Scale Datasets", in *proc. ICPR2008*, Dec. 2008. (CD-ROM).
10. C.C.Chang, C.J.Lin, "a library for support vector machines", `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, 2001.
11. `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#cod-rna`