Support Kernel Machine-Based Active Learning to Find Labels and a Proper Kernel Simultaneously

Yasusi Sinohara[1] and Atsuhiro Takasu[2]

[1] Central Research Institute of Electric Power Industry,
2-11-1 Iwado-kita, Komae-shi, Tokyo, 201-8511, Japan
sinohara@criepi.denken.or.jp
[2] National Institute of Informatics,
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
takasu@nii.ac.jp

Abstract. SVM-based active learning has been successfully applied when a large number of unlabeled samples are available but getting their labels is costly. However, the kernel used in SVM should be fixed properly before the active learning process. If the pre-selected kernel is inadequate for the target data, the learned SVM has poor performance. So, new active learning methods are required which effectively find an adequate kernel for the target data as well as the labels of unknown samples.
In this paper, we propose a two-phased SKM-based active learning method and a sampling strategy for the purpose. By experiments, we show that the proposed SKM-based active learning method has quick response suited to interaction with human experts and can find an appropriate kernel among linear combinations of given multiple kernels. We also show that with the proposed sampling strategy, it converges earlier to the proper combination of kernels than with the popular sampling strategy MARGIN.

1 Introduction

Active learning are used when a large number of unlabeled samples are available but getting their labels is costly, usually in cases that human experts assess the labels of unlabeled samples. Support vector machine (SVM)-based active learning has been successfully applied but the kernel used in SVM should be fixed properly in advance. If the pre-selected kernel is inadequate for the target data, the learned SVM has low predictive power. In batch learning, we can use time-consuming cross validation or other methods to find a proper kernel. But in active learning interacting with a human expert, the turnaround time, i.e., the time it takes to show an unlabeled sample for next labeling after one labeling, should be kept short. So a quickly responding active learning method is necessary which effectively finds an adequate kernel for the target data as well as the labels of unknown samples by interacting with experts.

In this paper, we propose a support kernel machine (SKM)-based active learner for the purpose. Because solving SKM is more time-consuming than SVM, we propose a two-phased SKM solver to reduce the turnaround time and also propose a sampling strategy SKM-SHIFT for SKM-based active learning.

2 Support Kernel Machines

Both SVM[1] and SKM[2] learn a separator $f(x)$ and predict the label $y \in \{\pm 1\}$ of input x by $y = sign(f(x))$.

For a given set of training samples $\{(x_i, y_i)\}_{i=1,\ldots,N}$ and a given (non-linear) feature mapping $\phi(x)$ or kernel $K(x,z) = \phi(x)^\top \phi(z)$, SVM finds a large margin separator $f(x) = w^\top \phi(x) + b$ by solving the following optimization problem.

$$\min_{\substack{w,b \\ \xi_i \geq 0}} \tfrac{1}{2}||w||^2 + C \textstyle\sum_i \xi_i \quad \text{subject to } y_i(w^\top \phi(x_i) + b) \geq 1 - \xi_i \tag{1}$$

The first term is for maximization of the margin $1/||w||$ and the second term is for the minimization of errors and the cost parameter C controls the trade-off between them. We shall henceforth refer to the optimal objective function value (1) as the error index which is closely related to the generalization error [1].

By duality, the following $\max_\alpha S(\alpha)$ equals to the error index.

$$\max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} \underbrace{\sum_i \alpha_i y_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j)}_{S(\alpha)} \tag{2}$$

$$\text{where } [A_i, B_i] = [\min(y_i C, 0), \max(y_i C, 0)]$$

By the optimal α^*, the optimal $w = \sum \alpha_i^* \phi(x_i)$ and $f(x) = \sum \alpha_i^* K(x_i, x) + b$. The samples whose $\alpha_i^* \neq 0$ are called support vectors.

In contrast to the SVM which uses a given single kernel $K(x_i, x_j)$, the SKM searches the SVM with the least error index whose kernel is a linear combination of given M kernels $\sum_{k=1}^M \beta_k K_k(x_i, x_j)$. Therefore SKM solves the following min-max or dual max-min problem [2].

$$\min_{\substack{\beta_k \geq 0 \\ \sum \beta_k = 1}} \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} S(\alpha; \beta) \tag{3}$$

$$= \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} \min_{\substack{\beta_k \geq 0 \\ \sum \beta_k = 1}} S(\alpha; \beta) \tag{4}$$

$$\text{where } S(\alpha; \beta) = \textstyle\sum_i \alpha_i y_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j \underbrace{\left(\displaystyle\sum_k \beta_k K_k(x_i, x_j) \right)}_{K(x_i, x_j; \beta)}$$

$$= \textstyle\sum_k \beta_k \underbrace{\left(\displaystyle\sum_i \alpha_i y_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K_k(x_i, x_j) \right)}_{S_k(\alpha)}$$

We refer $K(x_i, x_j; \beta)$ as a composite kernel and $K_k(x_i, x_j)$ as the k-th component kernel. $S_k(\alpha)$ is the objective function of the SVM with k-th component kernel.

SKM problem (3),(4) is an SVM problem w.r.t. α and a linear programming (LP) w.r.t. β, so the optimal α^*,β^* are sparse. The samples whose $\alpha_i^* \neq 0$ and the kernels whose $\beta_k^* > 0$ are called the support vectors and the support kernels respectively. The optimal $w = [\beta_1^* w_1, \cdots, \beta_M^* w_M]^\top$ where $w_k = \sum_i \alpha_i^* \phi_k(x_i)$ and $f(x) = \sum_k \beta_k^* \sum_i \alpha_i^* K_k(x_i, x) + b$. Thus, SKM can extract only critical samples and kernels for classification from given samples and component kernels.

SKM's error index $S(\alpha^*; \beta^*)$ equals to all support kernels' $S_k(\alpha^*)$ [3] and is smaller than any single kernel SVM's error index $\max_\alpha S_k(\alpha)$ because only weak duality ($\max_\alpha \min_k S_k(\alpha) \leq \min_k \max_\alpha S_k(\alpha)$) holds. Consequently SKM is expected to have higher precision than any single kernel SVM.

3 Two-phased SKM-based Active Learning: LASKM

SKM problem (3) is equivalent to the following semi-infinite LP (SKM-ILP).

$$\min_{\substack{\theta, \beta \geq 0 \\ \sum_k \beta_k = 1}} \theta \text{ s.t. } \textstyle\sum_k \beta_k S_k(\alpha) \leq \theta \text{ for all } \alpha \in \{\alpha | \alpha_i \in [A_i, B_i], \sum_i \alpha_i = 0\} \quad (5)$$

SKM-ILP can be solved by repeating the following steps starting from the initial constraint set $\mathcal{CS} = \{\beta_k \geq 0, \sum_k \beta_k = 1\}$ and some α^0,β^0 [3].

S1. find α^t s.t. $S(\alpha^t; \beta^{t-1}) > S(\alpha^{t-1}; \beta^{t-1})$.
S2. add constraint $\sum_k \beta_k S_k(\alpha^t) \leq \theta$ to the constraint set $\mathcal{CS}$ of LP.
S3. get β^t by solving the LP problem $\min_{(\theta,\beta) \in \mathcal{CS}} \theta$.

$\alpha^t = \text{argmax}\, S(\alpha; \beta^{t-1})$, the solution of SVM with kernel $K(\cdot, \cdot; \beta^{t-1})$, gives the tightest constraint at β^{t-1} in S2. However solving SVM is time-consuming and moreover it takes M times more time to solve $\max S(\alpha; \beta^{t-1})$ because the composite kernel has M component kernels. So the reduction of computation time of SVM is important to keep quick response during the active learning.

We therefore take a two-phased approach. During the active learning phase, we partially solve the SVM quickly using LASVM [4] and in the post-optimization phase, we completely optimize the SVM for all labeled data using a normal solver. The proposed two-phase active learner LASKM is shown in Algorithm 1.

LASVM is an efficient online SVM algorithm competitive with misclassification rates and out-speeding state-of-the-art SVM solvers. It is convenient to get α^t quickly in the active learning phase because it keep the set $\mathcal{S}$ of candidates of support vectors small by discarding blatant non-support vectors. It maintains three pieces of information: the set $\mathcal{S}$ and the coefficients α_i and the gradients $g_i = \frac{\partial}{\partial \alpha_i} S(\alpha) = y_i - \sum \alpha_j K(x_j, x_i)$ for $i \in \mathcal{S}$. Its building block operations are PROCESS and REPROCESS. $PROCESS(i)$ attempts to insert sample i into

[3] $S(\alpha^*; \beta^*) = \max_\alpha \min_k S_k(\alpha) = \min_k S_k(\alpha^*)$ because the inside of (4) is the LP-formulation of $\min_k S_k(\alpha)$. β^* can be determined because it satisfies $\sum \beta^* S_k(\alpha) \leq S(\alpha^*; \beta^*)$ for all α.

$\mathcal{S}$ and updates α_i and g_i for $i \in \mathcal{S}$. *REPROCESS* removes blatant non-support vectors from $\mathcal{S}$ and updates α_i, g_i for $i \in \mathcal{S}$.

In LASKM, we extend PROCESS and REPROCESS to $PROCESS(i, \beta)$ and $REPROCESS(\beta)$ which maintain the gradients of each component kernel $g_{[k]i} = y_i - \sum_j \alpha_j K_k(x_j, x_i)$ as well as that of the composite kernel g_i.

Algorithm 1 Two-Phased SKM-based Active Learner: LASKM

```
Require: Cost parameter $C$ and Component kernels $K_k(\cdot,\cdot)$, $k = 1, \cdots, M$
Require: samples $(x_i, y_i), i = 1, \cdots, N$ and Sampling Strategy $SS$
//Initialization
$t \leftarrow 0$.
constraint set $\mathcal{CS} \leftarrow \{\beta_k \geq 0, \sum \beta_k = 1\}$.
initial $(\alpha^0, g^0, g^0_{[k]}) \leftarrow (0, y, y)$, $\beta^0 \leftarrow (1/M, \cdots, 1/M)$.
Seed $L$ with a few samples of each class.
update $(\alpha^t, g^t, g^t_{[k]})$
//Active Learning Phase
repeat
    $t \leftarrow t + 1$
    Pick a sample $i_t$ by sampling strategy $SS$ and $L \leftarrow L \cup \{i_t\}$.
    update $(\alpha^t, g^t, g^t_{[k]})$ by PROCESS($i_t, \beta^{t-1}$) and REPROCESS($\beta^{t-1}$).
    $S^t_k \leftarrow S_k(\alpha^t), k = 1, \cdots, M$
    $CS \leftarrow CS \cup \{\sum_k \beta_k S^t_k \leq \theta\}$. //remove redundant constraints
    $(\theta^t, \beta^t) \leftarrow$ argmin$\{\theta | (\theta, \beta) \in CS\}$. //LP solution $\theta^t = S(\alpha^t; \beta^t)$.
    $g \leftarrow \sum_k \beta^t g^t_{[k]}$.
until $|L| \geq N$
//Post-Optimization Phase
repeat
    $t \leftarrow t + 1$
    get $(\alpha^t, g^t, g^t_k)$ by solving $\max S(\alpha; \beta^{t-1})$ using normal SVM algorithm.
    $S^t_k \leftarrow S_k(\alpha^t), k = 1, \cdots, M$ and $\theta^t_0 \leftarrow \sum \beta^{t-1}_k S^t_k$. //$\theta^t_0 = S(\alpha^t; \beta^{t-1})$
    $CS \leftarrow CS \cup \{\sum_k \beta_k S^t_k \leq \theta\}$. //remove redundant constraints
    $(\theta^t, \beta^t) \leftarrow$ argmin$\{\theta | (\theta, \beta) \in CS\}$. //LP solution $\theta^t = S(\alpha^t; \beta^t)$
    $g \leftarrow \sum_k \beta^t g^t_k$.
until $\theta^t_0 > 0$ and $|1 - \theta^t / \theta^t_0| \leq \epsilon$
```

4 Sampling Strategies for SKM-based Active Learning

The typical sampling strategies for SVM-based active learning are the followings:

1. RANDOM selects a sample randomly
2. MARGIN selects the sample nearest to the boundary ($\text{argmin}_x |f(x)|$).
3. KFF (Kernel Farthest First) selects the sample farthest to the boundary ($\text{argmax}_x |f(x)|$).
4. SHIFT selects the sample initially by KFF and by MARGIN depending on the stability of the model.

MARGIN strategy is known to be the most effective when the current model (α) is near optimal. However, when the current model is far from the optimal, true support vectors can exist far from the current boundary and exploration is more important than exploitation of the current boundary. SHIFT[5] initially uses KFF for exploration and shifts to MARGIN when the model becomes stable.

In case of SVM-based active learning, SHIFT's improved performance was limited on data sets that require extensive exploration such as checkerboard or COREL dataset, while remaining competitive on data sets that do not[5]. But SKM's model space (α, β) is larger than SVM's. So in SKM-based active learning, we think balancing exploration and exploitation is more important and propose to use an extended SHIFT strategy for the SKM-based active learning. The algorithm is shown in Algorithm 2. Along the line in [5], we defined the instability of SKM by the instability of angles between $w^t = (\beta_1 w_1^t, \cdots, \beta_M w_M^t)$ and w^{t-1}.

$$\mathrm{corr}(\alpha^t, \beta^t, \alpha^{t-1}, \beta^{t-1}) = \frac{\langle w^t, w^{t-1}\rangle}{\sqrt{\langle w^t, w^t\rangle\langle w^{t-1}, w^{t-1}\rangle}}$$

$$\text{where } \langle w^u, w^v\rangle = \sum_{i,j,k} \beta_k^u \beta_k^v \alpha_i^u \alpha_j^v K_k(x_i, x_j)$$

Algorithm 2 SKM-SHIFT(α, β, U)

```
Require: $U$:unlabeled samples, $O$:oracle
Require: $\lambda$, $\epsilon$, $A$:learning algorithm
if $U \neq \{\}$ return ({}, U) end if
$\phi \leftarrow \mathrm{corr}(\alpha, \beta, \alpha^0, \beta^0)$.
$p \leftarrow \max(\min(p_0 e^{-\lambda(\phi - \phi_0)}, 1 - \epsilon), \epsilon)$.
// probabilistic switching of strategy
$r \leftarrow$ random number generated uniformly between 0 and 1.
if $r < p$ then
    $x \leftarrow \mathrm{argmax}_{x \in U} \min_{i \in \mathcal{S}} K(x_i, x)$ //exploration: KFF(U)
else
    $x \leftarrow \mathrm{argmin}_{x \in U} |f(x)|$. //exploitation:MARGIN($U, \alpha, \beta$)
end if
$y \leftarrow O(x)$. //get the label of $x$ from the oracle
$(p_0, \alpha^0, \beta^0) \leftarrow (p, \alpha, \beta)$. //update state variables
return $((x, y), U - \{x\})$
```

When there are many unlabeled samples and we use all unlabeled samples as U in SKM-SHIFT/MARGIN/KFF, it takes much time to evaluate $|f(x)|$ or $K(x_i, x)$ of all x in U. To keep the turnaround short, we use a fixed number of randomly selected unlabeled samples as U. We set $|U| = 50$. This setting is practical because the probability that the maximum in U is over the 95th or 90th percentile of all is about 92% ($= 1 - 0.95^{50}$) or 99.5% ($= 1 - 0.90^{50}$).

MARGIN and KFF need the evaluation time proportional to $M \times |\mathcal{S}|$ to compute kernels between each sample in U and each sample in $\mathcal{S}$.

SKM-SHIFT needs additionally to compute $\mathrm{corr}(\alpha^t, \beta^t, \alpha^{t-1}, \beta^{t-1})$. However, because LASKM stores $\langle w^{t-1}, w^{t-1} \rangle$ and $g^t_{[k]j} = y_j - \sum_i \alpha^t_i K_k(x_i, x_j)$ in memory and $\langle w^t, w^s \rangle = \sum_{j,k} \alpha^s_j \beta^t_k \, \beta^s_k \, (\sum_i \alpha^t_i K_k(x_i, x_j))$, SKM-SHIFT needs to calculate $\sum_i \alpha^t_i K_k(x_i, x_j)$ only for the samples j which are the support vector candidates at time $t-1$ but excluded at time t (usually a few at most). So, SKM-SHIFT can select samples in almost equivalent time to MARGIN's.

5 Experiments

In the experiments, we use the USPS database [4] which contains 9298 handwritten digits (7329 for training, 1969 for testing). Each digit is a 16×16 image with zero mean and variance 1. We prepared 6 RBF kernels $\exp(-\gamma||x_i - x_j||^2)$ with $\gamma = 1, 2, 5, 10, 20, 50$ and make each single kernel SVM and the SKM using all 6 kernels learn the training data in batch or active learning. In batch learning, we use libsvm [6] for SVM and post-optimization part of LASKM for SKM. In active learning, we use LASVM and LASKM for SVM and SKM respectively. We set the cost parameter $C = 1000$ assuming the dataset is nearly separable, and tolerance $\tau = 1^{-3}$ (libsvm's default) for both SVM and SKM and the upper bound of $1 - \theta^t/\theta^t_0 = 1^{-6}$ for SKM. We use RANDOM, MARGIN, SKM-SHIFT as sampling strategies. KFF is excluded because it is obviously ineffective. The parameters in SKM-SHIFT(Algorithm 2) are $\lambda = 0.5$, $\phi_0 = 0.3$, $\epsilon = 0.05$ [5] and the size of random sampling is $|U| = 50$.

Usually assessing labels of over 1000 samples are very stressful. So we set the maximum number of learned samples to 900 in active learning. The analysis of changes of turnaround time, precision etc. are based on the data when the numbers of learned samples are 12, 14, …, 28, 30, 40, 50, …, 90, 100, 200,…, 800, 900.

We made 20 trials for each experimental setting. In each trial, we randomly select a pair of one positive and one negative sample as the initial sampling data.

We use a Pentium-4 2.6GHz Windows XP machine for the experiments.

5.1 Comparison between single kernel SVMs and SKM

We first compared the optimal single kernel SVMs and the optimal SKM when learning all training data in batch.

Table.1 lists the error index ($\max_\alpha S_k(\alpha)$ of SVMs and $S(\alpha^*; \beta^*)$ of SKM) and precision against the test data for digit '0'.

SKM's error index is the lowest among all and the optimal weights β^*_γ of the support kernels are 0.5 (γ=5), 0.2 (γ=10), 0.3 (γ=20) and those of the non-support kernels (γ=1, 50) are 0. SKM successfully selected the three kernels with

[4] ftp://ftp.kyb.tuebingen.mpg.de/pub/bs/data/

[5] λ, ϕ_0,ϵ are not optimized. we set them referring to [5].

Table 1. Error Index and Precision for ’0’

γ	Error Index	Precision
1	1334.1	99.44%
2	467.6	99.49%
5	175.6	99.54%
10	163.3	99.64%
20	331.4	98.93%
50	1082.4	93.30%
SKM	149.8	99.54%

Table 2. β^* selected by SKM

digit	RBF parameterγ					
	1	2	5	10	20	50
0	0.00	0.00	0.52	0.19	0.29	0.00
1	0.00	0.00	0.48	0.24	0.22	0.06
2	0.00	0.00	0.48	0.28	0.13	0.11
3	0.00	0.00	0.53	0.17	0.29	0.01
4	0.00	0.00	0.44	0.36	0.11	0.09
5	0.00	0.00	0.46	0.29	0.15	0.10
6	0.00	0.00	0.47	0.21	0.32	0.00
7	0.00	0.00	0.52	0.20	0.24	0.04
8	0.00	0.00	0.44	0.37	0.03	0.16
9	0.00	0.00	0.42	0.38	0.16	0.05

low error index as the support kernels and composed a composite kernel with a lower error index. The precision of the optimal SKM is 99.54% which is the second best behind the SVM’s with $\gamma = 10$ (99.64%) and the SMV’s with $\gamma = 50$ is the worst (93.30%). For other digits, SKMs also had the least error index and the best or second best precision compared with the single kernel SVMs [6].

In active learning, this feature is a great advantage of SKM against SVM. Because the user doesn’t know the labels of data when selecting kernels, there are relatively high risks that the user selects an inefficient kernel (such as $\gamma = 50$ for digit ’0’) in SVM-based active learning but in SKM-based active learning, we can reach to the solution comparable to the best single kernel SVM’s starting from a set of candidate kernels.

Table. 2 shows the optimal β^* of SKM for each digit. For all digits, the SKM’s major support kernels are γ =5, 10, 20 and γ =5 has the highest weight. However, the SVM with the least error index among the single kernel SVMs is γ =10 and not γ =5. In the optimal SKM, the RBF kernel with $\gamma = 5$ gives a rough shape of the decision boundary and the RBFs with $\gamma = 10$ or 20 refine it.

5.2 Comparison of Sampling Strategy in SKM-based active learning

Turnaround Time Figure. 1 shows the average turnaround time (dotted line) and the average number of support vector candidates $|\mathcal{S}|$ (solid line). The numbers of learned samples are on the horizontal axis.

RANDOM responses very rapidly (0.02 sec. at 900 samples) but it fails to select proper support vector candidates effectively. So the number of candidates stays 200, the one third of those of the other two strategies at 900 samples.

MARGIN and SHIFT have similar turnaround proportional to the number of support vector candidates. Even at 900 samples, it responses within one second (0.9 sec.).

[6] The error index is only an “estimate” of generalization error. So the precision of the optimal SKM is comparable to the best SVM but can be inferior to it in some cases.

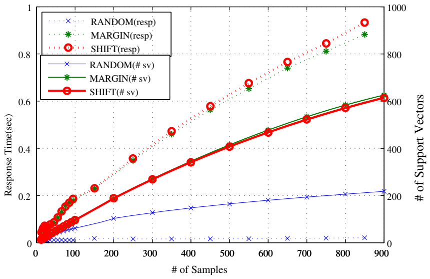

Fig. 1. Turnaround time and the number of SVs of LASKM

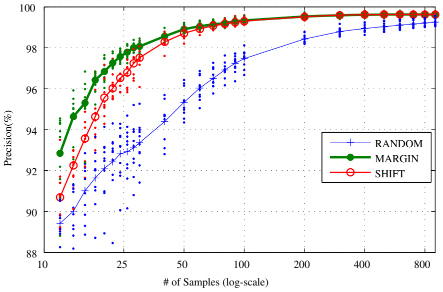

Fig. 2. Transition of Precision

As for the simple active SKM learner, which uses MARGIN and incrementally optimizes SKM completely at each sampling, its average turnaround was 2.5 sec. for 100–200 samples, 10 sec. for 200–300 samples, 15 sec. for 300–400 samples. These responses are too slow for comfortable interaction with human experts. In contrast, the two-phased active learner LASKM has very quick response suitable to support interactive active learning.

Temporal Transition of Precision We show the changes of precision averaged over 200 trials (20 trials of ten digits) for each strategy in Figure 2 because the performance are very similar regardless of digits.

RANDOM is obviously poor and MARGIN and SKM-SHIFT have similar performance in precision. More precisely, MARGIN is slightly better than SKM-SHIFT when the number of learned samples is small (less than 200 for digit '3' and '4' and 30 to 80 for other digits) but the differences of precision between MARGIN and SKM-SHIFT are less than 0.5% even at 50 samples for all digits.

Concerning precision, SKM-SHIFT didn't make much difference with MARGIN just like the SHIFT in the SVM-active learning as described in [5].

Temporal Transition of Kernel Weight Concerning the changes of kernel weight β^t, the performances of MARGIN and SKM-SHIFT are different.

Figure 3 shows the temporal changes of kernel weights β^t (averaged over 20 trials) of digit '0'. The optimal $\beta^* = (0, 0.5, 0.2, 0.3, 0)$ is shown in the right most in Figure3. MARGIN gives more weight to $\gamma = 10$ until the number of samples reaches about 200 and requires over 400 samples to reach β^*. On the other hand, SHIFT reached β^* at about 200 samples.

The same tendency appears in other digits. MARGIN is likely to select $\gamma = 10$ which has the least error index in single kernel SVMs in the earlier stage ($\beta_{\gamma=10}$ is around 0.5 or 0.6 at 20 samples). SHIFT is likely to reach to β^* earlier than MARGIN.

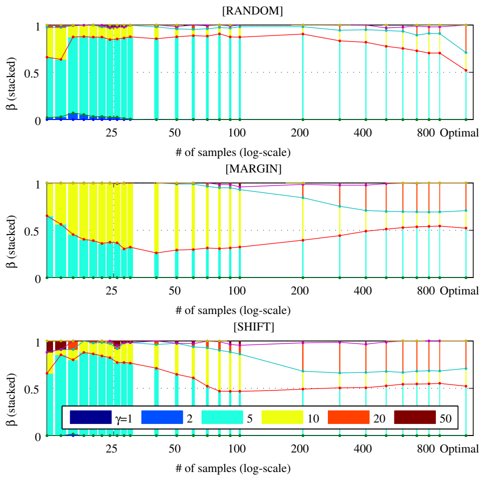

Fig. 3. Transition of β

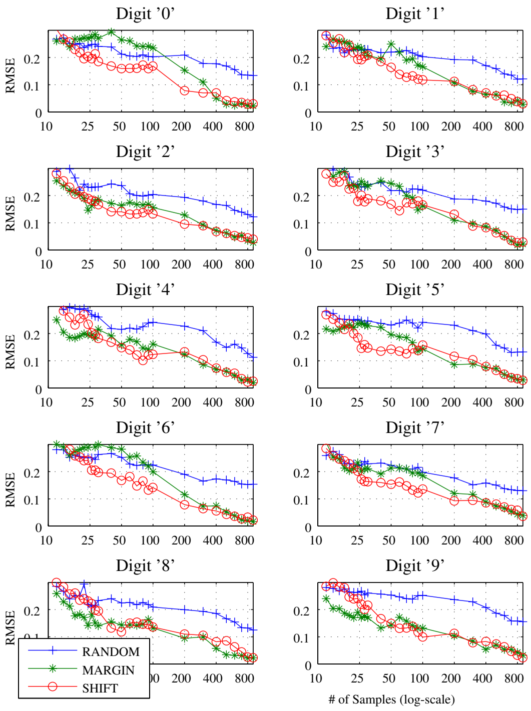

Fig. 4. Transition of Errors of β, $||\beta - \beta^*||$

Figure 4 shows the changes of $\|\beta^t - \beta^*\|$, the estimation error of β^*. Especially, SHIFT's β^t are closer to β^* than MARGIN until 100 samples except for digits '4', '8' and '9' (at significance level 90%).

The reason of the exceptional good performance of MARGIN for digit '4', '8', '9' can be considered as follows. MARGIN is likely to select $\gamma = 10$ in initial phase as mentioned above and digits '4', '8', '9' have larger optimal weight $\beta^*_{\gamma=10}$ than other digits as shown in Table 2. So, the initial decision boundary by MARGIN is likely to be closer to the optimal boundary and this causes the good estimate of β^* at earlier stages by MARGIN.

But in general, the best kernel for single kernel SVM is different from the kernel having large weight in SKM. So, SKM-SHIFT can be considered as more robust and better estimator of optimal kernel weight β^* than MARGIN.

6 Conclusions and Future Works

In this paper, we propose the SKM-based active learning and, for the purpose, a two-phased algorithm LASKM and a sampling strategy SKM-SHIFT based on SHFIT strategy whose improvement was limited in case of SVM-based learning.

By experiments, we show that the proposed LASKM has quick response necessary for interactive active learning and can find an appropriate composite kernel among combinations of given component kernels which is comparable to the best component kernel with respect to the predictive power.

We also show that with the proposed sampling strategy SKM-SHIFT it converges earlier to the optimal combination β^* of composite kernel than with the popular sampling strategy MARGIN with some exception, while it remains comparable with MARGIN concerning precision of prediction of labels.

We are now conducting empirical evaluations of the LASKM for different datasets and are also planning detailed sensitivity analyses of SKM-shift's parameters.

References

1. Vapnik, V.: Statistical Learning Theory. John Wiley and Sons Inc.(1998)
2. Bach, F.R., Lanckriet, G.R.G., Jordan, M.I.: Multiple Kernel Learning, Conic Duality, and the SMO Algorithm. Proc. ICML, Banff Canada (2004)
3. Sonnenburg, S., Ratsch, G., and Schafer, C.: A general and efficient multiple kernel learning algorithm. Advances in Neural Information Processing Systems 18, . MIT Press, Cambridge MA (2006)
4. Bordes, A., Ertekin, S., Weston, J., Bottou, L.: Fast Kernel Classifiers with Online and Active Learning. Journal of Machine Learning Research 6 (2005)1579–161
5. Osugi, T., Kun. D., Scott, S.: Balancing Exploration and Exploitation : A New Algorithm for Active Machine Learning. Proc. ICDM 05 (2005) 330–337
6. Chang, C.-C. and Lin, C.-J.: LIBSVM: a library for support vector machines. (2001) Software available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm.