Multi-task Learning with Operator-valued Reproducing Kernels

Asahi Ushio and Masahiro Yukawa
Dept. Electronics and Electrical Engineering, Keio University, Japan.
E-mail: ushio@ykw.elec.keio.ac.jp, yukawa@elec.keio.ac.jp

Abstract

We present a novel adaptive multi-task learning framework to estimate nonlinear vector-valued functions. The presented framework is based on operator-valued reproducing kernel Hilbert spaces, and is basically an extension of a kernel adaptive filtering algorithm called the hyperplane projection along affine subspace (HYPASS). The efficacy of the approach is shown by numerical examples.

Nomenclature

$\boldsymbol{u}^{\mathsf{T}}$	: Transpose of vector $\boldsymbol{u}$
$\mathbb{R}^{a\times b}$	: the space of real $a \times b$ matrices
$\mathbb{N}$	: the set of all positive integers
$\mathcal{U}$	: a set
$\mathcal{D}$	: a Hilbert space with inner product $\langle\cdot,\cdot\rangle_{\mathcal{D}}$
$B(\mathcal{D})$	: the set of all bounded linear operator from $\mathcal{D}$ to itself.
$\mathcal{H}$	: a Hilbert space on $\mathcal{U}$ with values in $\mathcal{D}$ with inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$
$B_+(\mathcal{D})$	: $\{A \in B(\mathcal{D}) \mid d \in \mathcal{D}, \langle d, Ay\rangle_{\mathcal{D}} \geq 0\}$
κ	: a reproducing kernel

1 Introduction

We encounter nonlinear vector-valued functions in a wide range of learning problems such as multi-task learning. In multi-task learning, the problem is to estimate nonlinear vector-valued functions that are related to each other. A vector-valued function can be regarded as a set of scalar-valued functions, each of which is what to be estimated in each task. The point here is to exploit the correlation among the tasks for enhancing the learning efficiency. In [1,2], multi-task learning has been studied in batch scenarios within the framework of reproducing kernel Hilbert spaces (RKHSs). In [3], to reduce the computational complexity, Audiffren and Kadri have proposed an online method by extending the naive online R_{reg} minimization algorithm (NORMA) [4] to vector-valued functions. The method in [3], called the operator NORMA (ONORMA), is based on the framework of operator-valued RKHS [5]. Since ONORMA employs no novelty criterion to build an efficient dictionary and adopts a simple truncation rule to limit the dictionary size, a more efficient scheme is required to enhance practical applicability.

In this paper, we propose an efficient online multi-task learning algorithm based on iterative orthogonal projections in a operator-valued RKHS. The proposed algorithm is derived by extending the idea of the hyperplane projection along affine subspace (HYPASS) [6,7], thus named the Operator HYPASS (OHYPASS) algorithm. The proposed algorithm selectively builds a dictionary based on Platt's criterion, and measurements that are not added to the dictionary are still used to polish the coefficients by means of the projection along an affine subspace. Numerical examples show the efficacy of the proposed scheme.

2 Operator-valued Kernel

We introduce the idea of operator-valued RKHS given in [5]. To establish operator-valued RKHS, define a RKHS as below.

Definition (Reproducing kernel Hilbert space) $\mathcal{H}$ is a RKHS if, for any $d \in \mathcal{D}$, $u \in \mathcal{U}$ and $f \in \mathcal{H}$, the linear functional (evaluation functional)

$$\mathcal{F}_{u,d}(f) = \langle f(u), d\rangle_{\mathcal{D}} \qquad (1)$$

is continuous.

By the Riesz Lemma, there exists a function $\rho : \mathcal{U} \times \mathcal{D} \to \mathcal{H}$ such that

$$\mathcal{F}_{u,d}(f) = \langle \rho(u,d), f\rangle_{\mathcal{H}}, \quad \forall f \in \mathcal{H}. \qquad (2)$$

From (1), (2), we attain

$$\langle f(u), d\rangle_{\mathcal{D}} = \langle \rho(u,d), f\rangle_{\mathcal{H}}. \qquad (3)$$

As $\rho(u,d)$ is linear in $\mathcal{D}$, it can be written as the linear operator $\kappa(u,u^*) : \mathcal{D} \to \mathcal{D}$ s.t.

$$\kappa(u,u^*)d := \rho(u,d)u^*, \quad \forall d \subset \mathcal{D}, \forall u, u^* \subset \mathcal{U}. \qquad (4)$$

From (3) and (4), we attain a reproducing property s.t.

$$\langle f(u), d\rangle_{\mathcal{D}} = \langle \kappa(u,\cdot)d, f\rangle_{\mathcal{H}}. \qquad (5)$$

Then we call the operator-valued mapping $\kappa(\cdot,\cdot) : \mathcal{U} \times \mathcal{U} \to B(\mathcal{D})$ defined by (4) the reproducing kernel of $\mathcal{H}$. $\kappa(\cdot,\cdot)$ is illustrated in Fig. 1.

3 Operator HYPASS Algorithm

3.1 Notation

We define $\boldsymbol{U}_n = [\boldsymbol{u}_{1,n}, \ldots, \boldsymbol{u}_{\tau,n}] \in \mathcal{U} = \mathbb{R}^{L\times\tau}$, $\boldsymbol{u}_{t,n} = [u_{t,n,1}, \ldots, u_{t,n,L}]^{\mathsf{T}} \in \mathbb{R}^L$ as the input space and $\boldsymbol{d}_n = [d_{1,n}, \ldots, d_{\tau,n}]^{\mathsf{T}} \in \mathcal{D} = \mathbb{R}^{\tau}$ as the output space at sample index n. Here, an operator $F_n \in \mathcal{H}$ is given as

$$\boldsymbol{d}_n = F_n(\boldsymbol{U}_n) = [f_{1,n}(\boldsymbol{U}_n), \ldots, f_{\tau,n}(\boldsymbol{U}_n)]^{\mathsf{T}} \qquad (6)$$

where $f_{t,n}(\boldsymbol{U}_n) \in \mathbb{R}$ is the tth task for $t = 1, \ldots, \tau$.

3.2 Proposed Algorithm

We present an operator-valued kernel approach to estimate the optimal solution. An example of operator-

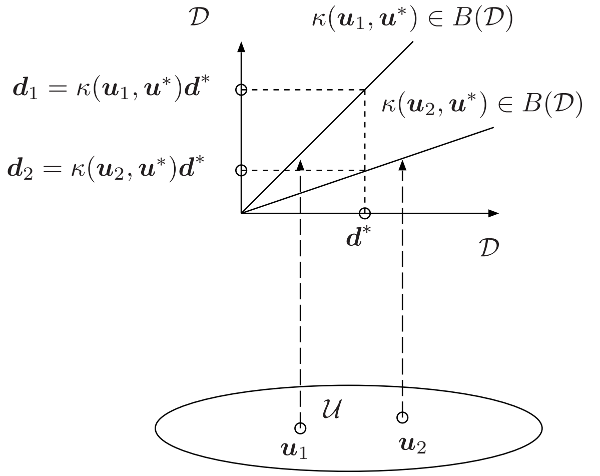
$\mathcal{D}$
$\kappa(\boldsymbol{u}_1, \boldsymbol{u}^*) \in B(\mathcal{D})$
$\boldsymbol{d}_1 = \kappa(\boldsymbol{u}_1, \boldsymbol{u}^*)\boldsymbol{d}^*$
$\kappa(\boldsymbol{u}_2, \boldsymbol{u}^*) \in B(\mathcal{D})$
$\boldsymbol{d}_2 = \kappa(\boldsymbol{u}_2, \boldsymbol{u}^*)\boldsymbol{d}^*$
$\boldsymbol{d}^*$
$\mathcal{D}$
$\mathcal{U}$
$\boldsymbol{u}_1$
$\boldsymbol{u}_2$

Figure 1: Operator-valued reproducing kernel $\kappa(\cdot, \boldsymbol{u}^*)$ with $\boldsymbol{u}^* \in \mathcal{U}$ fixed can be regarded as a mapping from $\mathcal{U}$ to the operator space $\mathcal{H}$.

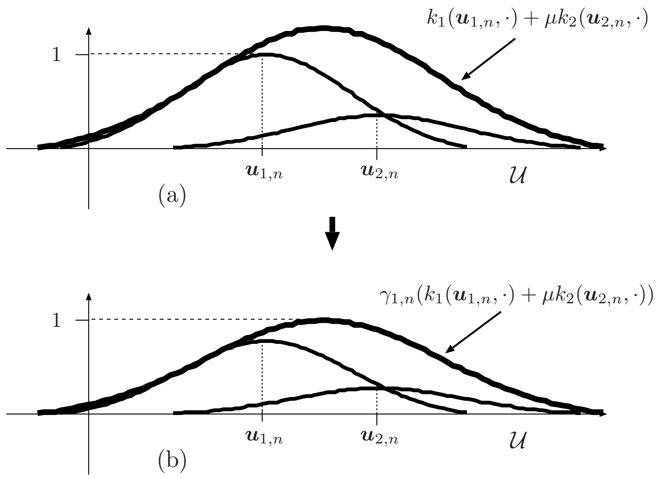

Figure 2: Illustration of $K_t(\boldsymbol{U}_n, \cdot)$ in two-task case. (a) Scale $k_2(\boldsymbol{u}_{2,n}, \cdot)$ down by μ, and then (b) normalize the superposition $k_1(\boldsymbol{u}_{1,n}, \cdot) + \mu k_2(\boldsymbol{u}_{2,n}, \cdot)$ by $\gamma_{1,n}$.

valued kernel is given as

$$\kappa(\boldsymbol{U}, \boldsymbol{V}) := \begin{bmatrix} k_1(\boldsymbol{u}_1, \boldsymbol{v}_1) & \mu k_1(\boldsymbol{u}_1, \boldsymbol{v}_2) & \cdots & \mu k_1(\boldsymbol{u}_1, \boldsymbol{v}_\tau) \\ \mu k_2(\boldsymbol{u}_2, \boldsymbol{v}_1) & k_2(\boldsymbol{u}_2, \boldsymbol{v}_2) & \cdots & \mu k_2(\boldsymbol{u}_2, \boldsymbol{v}_\tau) \\ \vdots & \vdots & \ddots & \vdots \\ \mu k_\tau(\boldsymbol{u}_\tau, \boldsymbol{v}_1) & \mu k_\tau(\boldsymbol{u}_\tau, \boldsymbol{v}_2) & \cdots & k_\tau(\boldsymbol{u}_\tau, \boldsymbol{v}_\tau) \end{bmatrix} \tag{7}$$

for $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_\tau], \boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_\tau] \in \mathcal{U}$ where $k_t(\cdot, \cdot)$ is the reproducing kernel of scalar-valued RKHS $\mathcal{H}_t : \mathbb{R}^L \to \mathbb{R}$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_t}$ on $\mathcal{U}$ and $\mu \in [0, 1]$ is the parameter governing the interactions among tasks. In the extreme case, $\mu = 0$ gives no iteraction among tasks and $\mu = 1$ gives the highest interactions. The linear variety that enforces the instantaneous errors for all tasks to be zero simultaneously is defined as

$$\begin{aligned} \mathcal{V}_n &:= \{F \in \mathcal{H} | F(\boldsymbol{U}_n) = \boldsymbol{d}_n\} \\ &= \{F \in \mathcal{H} | \langle F(\boldsymbol{U}_n), \boldsymbol{e}_t \rangle_{\mathcal{D}} = \langle \boldsymbol{d}_n, \boldsymbol{e}_t \rangle_{\mathcal{D}} = d_{t,n}, \forall t\} \\ &= \{F \in \mathcal{H} | \langle \kappa(\boldsymbol{U}_n, \cdot)\boldsymbol{e}_t, F \rangle_{\mathcal{H}} = d_{t,n}, \forall t\}, \end{aligned}$$

where $\boldsymbol{e}_t$ is the tth standard orthonormal basis vector of $\mathcal{D}$ for $t = 1, \ldots, \tau$. Here the last equality is verified by the reproducing property (5). Since $f_{t,n} \in \mathcal{H}_t$ in (6) the RKHS $\mathcal{H}$ associated with the kernel defined in (7) can be regarded as the product space

$$\mathcal{H} = \mathcal{H}_1 \times \cdots \times \mathcal{H}_\tau.$$

Here, we have

$$\begin{aligned} &\kappa(\boldsymbol{U}_n, \cdot)\boldsymbol{e}_t = \\ &[\mu k_1(\boldsymbol{u}_1, \cdot), \mu k_2(\boldsymbol{u}_2, \cdot), \ldots, \underbrace{k_t(\boldsymbol{u}_t, \cdot)}_{t\text{th}}, \ldots, \\ &\mu k_{\tau-1}(\boldsymbol{u}_{\tau-1}, \cdot), \mu k_\tau(\boldsymbol{u}_\tau, \cdot)]^\mathsf{T} \\ &\langle \kappa(\boldsymbol{U}_n, \cdot)\boldsymbol{e}_t, F \rangle_{\mathcal{H}} = \\ &\sum_{a=1}^{\tau} \mu \langle k_a(\boldsymbol{u}_{a,n}, \cdot), f_a \rangle_{\mathcal{H}_a} + (1 - \mu)\langle k_t(\boldsymbol{u}_{t,n}, \cdot), f_t \rangle_{\mathcal{H}_t}. \end{aligned}$$

Here the dictionary subspace at n is given by

$$\begin{aligned} \mathcal{M}_n &:= \mathcal{M}_{1,n} \times \cdots \times \mathcal{M}_{\tau,n} \\ \mathcal{M}_{t,n} &:= \operatorname{span}\{K_t(\boldsymbol{U}_i, \cdot)\}_{i \in \mathcal{J}_n^t} \end{aligned} \tag{8}$$

where

$$K_t(\boldsymbol{U}_i, \cdot) := \gamma_{t,i} \left[(1 - \mu)k_t(\boldsymbol{u}_{t,i}, \cdot) + \mu \sum_{a=1}^{\tau} k_a(\boldsymbol{u}_{a,i}, \cdot) \right]$$

with the normalization factor

$$\gamma_{t,i} := \frac{1}{\mu \sum_{a=1}^{\tau} k_a(\boldsymbol{u}_{a,i}, \boldsymbol{u}_{t,i}) + (1 - \mu)k_t(\boldsymbol{u}_{t,i}, \boldsymbol{u}_{t,i})}$$

and $\mathcal{J}_n^t = \{j_1^{(t,n)}, \ldots, j_{r_n^t}^{(t,n)}\} \subset \{0, 1, \ldots, n-1\}$ is the tth dictionary index set at n (r_n^t is the dictionary size). A simple case of two tasks with Gaussian kernels is illustrated in Fig. 2.

To enhance the adaptivity and reduce the dictionary size efficiently, we exploit the idea of HYPASS [6,7]. The estimate F_n^* is updated as

$$F_{n+1}^* = F_n^* + \lambda(P_{\mathcal{V}_n \cap \mathcal{M}_n}(F_n^*) - F_n^*), \tag{9}$$

where $\lambda \in [0, 2]$ is the step size and

$$P_{\mathcal{V}_n \cap \mathcal{M}_n}(F_n^*) = F_n^* + \sum_{t=1}^{\tau} \boldsymbol{e}_t \beta_{t,n} P_{\mathcal{M}_{t,n}}(K_t(\boldsymbol{U}_n, \cdot))$$

is the orthogonal projection onto the intersection of $\mathcal{M}_n$ and $\mathcal{V}_n$. Here,

$$\beta_{t,n} := \frac{d_{t,n} - f_{t,n}^*(\boldsymbol{U}_n)}{\sum_{i=1}^{r_n^t} \alpha_{t,i} K_t(\boldsymbol{U}_{j_i^{(t,n)}}, \boldsymbol{U}_n)},$$

and

$$P_{\mathcal{M}_{t,n}}(K_t(\boldsymbol{U}_n, \cdot)) = \sum_{i=1}^{r_n^t} \alpha_{t,i} K_t(\boldsymbol{U}_{j_i^{(t,n)}}, \cdot)$$

is the projection onto $\mathcal{M}_{t,n}$ with $\boldsymbol{\alpha}_{t,n} = [\alpha_{t,1}, \ldots, \alpha_{t,r_n^t}]^\mathsf{T} \in \mathbb{R}^{r_n^t}$ characterized by the following normal equation:

$$\boldsymbol{K}_{t,n}\boldsymbol{\alpha}_{t,n} = \boldsymbol{p}_{t,n},$$

where

$$\boldsymbol{K}_{t,n} := \begin{bmatrix} K_t(\boldsymbol{U}_{j_1^{(n)}}, \boldsymbol{U}_{j_1^{(n)}}) & \dots & K_t(\boldsymbol{U}_{j_1^{(n)}}, \boldsymbol{U}_{j_{r_n^t}^{(n)}}) \\ \vdots & \ddots & \vdots \\ K_t(\boldsymbol{U}_{j_1^{(n)}}, \boldsymbol{U}_{j_{r_n^t}^{(n)}}) & \dots & K_t(\boldsymbol{U}_{j_{r_n^t}^{(n)}}, \boldsymbol{U}_{j_{r_n^t}^{(n)}}) \end{bmatrix} \in \mathbb{R}^{r_n^t \times r_n^t}$$

$$\boldsymbol{p}_{t,n} := [K_t(\boldsymbol{U}_{j_1^{(n)}}, \boldsymbol{U}_n), \dots, K_t(\boldsymbol{U}_{j_{r_n^t}^{(n)}}, \boldsymbol{U}_n)]^\mathsf{T} \in \mathbb{R}^{r_n^t}.$$

3.3 Relation to prior works and possible extension

Kernel-based methods to estimate vector-valued functions have been proposed in [1–3, 8], among which the studies in [3, 8] consider online scenarios. The online algorithm in [8], which is basically an extension of [9], has been derived in a context different from multi-task learning. Hence, it does not take into account the correlation among tasks. The online algorithms in [3] build a dictionary using all the input vectors and discard the oldest datum by a simple truncation rule in a manner analogous to NORMA. This strategy has two limitations: (i) the dictionary could become highly redundant and (ii) an important atom of the dictionary would be discarded. In [7], some ideas to overcome the limitations have been proposed and the proposed scheme extends those ideas basically. (We adopt Platt's criterion [10] to build an efficient dictionary.) We also adopt the selective updating strategy presented in [7], updating a fixed number, say Q, of coefficients at each iteration. Another major difference is that a single-valued function space over $\mathbb{R}^{L\tau}$ is used to define an operator valued kernel in [3] while τ single-valued function spaces over $\mathbb{R}^L$ are used in the present work.

In [3], a multikernel algorithm has also been presented. The proposed scheme can also be extended by using the framework of multikernel adaptive filtering [11, 12], as will be investigated in our future works.

3.4 Computational complexity

We consider the complexity in terms of the number of multiplications involved in the filter update per task. The complexity of OHYPASS is $(Q^2-Q)L\tau/2+O(Q^3)+Q^2+2Q+\tau$. Here, the complexity for the kernel evaluation is $(Q^2-Q)L/2$, that for the inversion of $\boldsymbol{K}_{t,n}$ is $O(Q^3)$, that for a matrix-vector multiplication is Q^2, and that for the calculation of $\beta_{t,n}$ is $2Q$. The complexity of HYPASS is $(Q^2-Q)L/2+O(Q^3)+Q^2+2Q$. Compared to HYPASS, the kernel evaluation of OHYPASS requires τ times more multiplications and each $\gamma_{t,i}$ requires L multiplication. Since OHYPASS exploits the inputs of all tasks, the complexity of OHYPASS increases linearly in τ. For instance, in the numerical example, we use $\tau = 5$, $L = 1$, and $Q = 1$, which makes the difference of the complexities be no more than 25 per iteration.

4 Numerical Examples

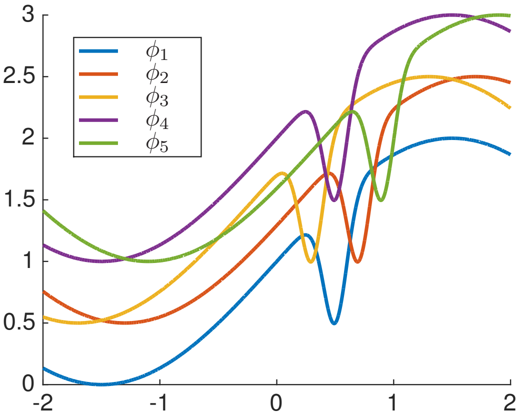

Figure 3: The functions $\phi_1, \dots, \phi_5$ to be estimated.

We show the efficacy of the proposed algorithm for a toy example. We consider the following model with $L = 1, \tau = 5$: $\Phi(\boldsymbol{U}_n) = [\phi_1(u_{1,n}), \dots, \phi_5(u_{5,n})]^\mathsf{T} \in \mathcal{D} := \mathbb{R}^5$ for $\boldsymbol{U}_n = [u_{1,n}, \dots, u_{5,n}] \in \mathcal{U} = [-2, 2]^5 \subset \mathbb{R}^{1\times 5}$, $\phi_t(u_{t,n}) = \phi(u_{t,n} - \theta_u^{(t)}) + \theta_d^{(t)}$, $\phi(u) := 1 + \sin\left(\frac{\pi}{3}(u)\right) - \exp\left(-\frac{(u-0.5)^2}{2\times 0.1^2}\right)$,

$$[\theta_d^{(1)}, \theta_d^{(2)}, \theta_d^{(3)}, \theta_d^{(4)}, \theta_d^{(5)}] = [0, 0.5, 0.5, 1, 1]$$

$$[\theta_u^{(1)}, \theta_u^{(2)}, \theta_u^{(3)}, \theta_u^{(4)}, \theta_u^{(5)}] = [0, 0.2, -0.2, 0, 0.4].$$

The function ϕ_t is associated with task t and Fig. 3 shows those functions. The input sequence $(\boldsymbol{U}_n)_{n\in\mathbb{N}}$ is randomly drawn from the uniform distribution over the input space $\mathcal{U}$. The observed data is given by $\boldsymbol{d}_n := \Phi(\boldsymbol{U}_n) + \boldsymbol{\nu}_n$ with zero mean i.i.d. Gaussian noise $\boldsymbol{\nu}_n \in \mathbb{R}^5 \sim \mathcal{N}([0,0,0,0,0]^\mathsf{T}, 2\times 10^{-3}\boldsymbol{I})$ where $\boldsymbol{I} \in \mathbb{R}^{5\times 5}$ is the identity matrix. We compare the performance of the proposed algorithm with HYPASS applied to each task independently. For fairness, both algorithms use Platt's criterion for the dictionary construction. We set $Q = 1$ for both algorithms. We test 300 independent trials and compute the average mean squared error (MSE) for 5 tasks. We use normalized Gaussian kernel $k(u_1, u_2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(u_1-u_2)^2}{2\times\sigma^2}\right)$, $u_1, u_2 \in \mathbb{R}$ with $\sigma = 0.2$ for all tasks, $\mu = 0.2$, $\lambda = 0.1$. According to Platt's criterion [10], $k_t(u_{t,n}, \cdot)$ is regarded to be novel if

$$\max_{i\in\mathcal{J}_{n-1}^t} \exp\left(-\frac{(u_{t,i}-u_{t,n})^2}{2\sigma}\right) < \delta \qquad (10)$$

and if $|d_{t,n} - f_{t,n}(u_{t,n})|^2 > \epsilon |f_{t,n}(u_{t,n})|^2$ for $\delta := 0.9$, $\epsilon := 10^{-4}$.

Fig. 4 shows the MSE learning curves that are avaraged over time with the window size 20 for visual clarity. It is seen that the proposed algorithm exhibits faster initial convergence compared to HYPASS. This is due to the effective use of the correlation among tasks.

5 Conclusion

We presented an efficient adaptive multi-task learning framework to estimate nonlinear vector-valued functions based on operator-valued reproducing kernel Hilbert spaces. The presented scheme was basically an extension of HYPASS algorithm. Numerical examples showed that

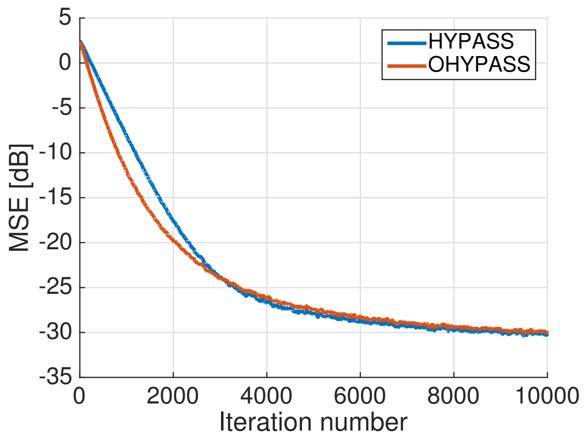

Figure 4: MSE learning curves

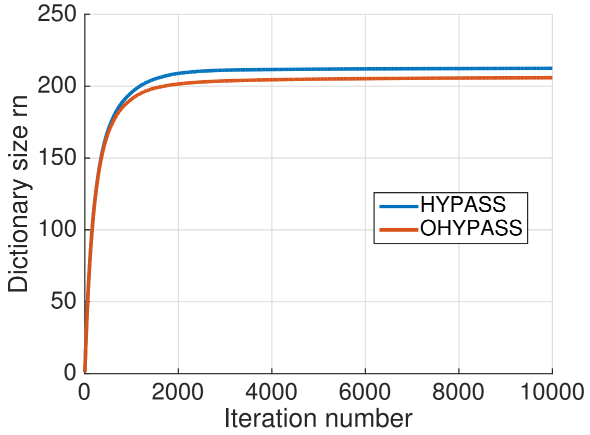

Figure 5: Dictionary evolutions

the proposed multi-task learning scheme attained faster initial convergence than HYPASS applied separately to each task.

Acknowledgment

This work was partially supported by JSPS Grants-in-Aid (24760292,15K06081).

References

[1] Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil, “Learning multiple tasks with kernel methods,” in *Journal of Machine Learning Research*, 2005, pp. 615–637.

[2] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil, “Convex multi-task feature learning,” *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.

[3] Julien Audiffren and Hachem Kadri, “Online learning with multiple operator-valued kernels,” *arXiv preprint arXiv:1311.0222*, 2013.

[4] Jyrki Kivinen, Alexander J Smola, and Robert C Williamson, “Online learning with kernels,” *Signal Processing, IEEE Transactions on*, vol. 52, no. 8, pp. 2165–2176, 2004.

[5] Charles A Micchelli and Massimiliano Pontil, “On learning vector-valued functions,” *Neural computation*, vol. 17, no. 1, pp. 177–204, 2005.

[6] Masahiro Yukawa and Ryu ichiro Ishii, “An efficient kernel adaptive filtering algorithm using hyperplane projection along affine subspace,” *in Proceedings of EUSIPCO*, pp. 2183–2187, 2012.

[7] M. Takizawa and M. Yukawa, “Adaptive nonlinear estimation based on parallel projection along affine subspaces in reproducing kernel Hilbert space”,” *IEEE Trans. Signal Processing*, vol. 63, no. 16, pp. 4257–4269, Aug. 2015.

[8] Felipe Tobar, Sun-Yuan Kung, Danilo P Mandic, et al., “Multikernel least mean square algorithm,” *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 25, no. 2, pp. 265–277, 2014.

[9] Cédric Richard, José Carlos M Bermudez, and Paul Honeine, “Online prediction of time series data with kernels,” *Signal Processing, IEEE Transactions on*, vol. 57, no. 3, pp. 1058–1067, 2009.

[10] John Platt, “A resource-allocating network for function interpolation,” *Neural computation*, vol. 3, no. 2, pp. 213–225, 1991.

[11] M. Yukawa, “Multikernel adaptive filtering,” *IEEE Trans. Signal Processing*, vol. 60, no. 9, pp. 4672–4682, Sept. 2012.

[12] M. Yukawa, “Adaptive learning in Cartesian product of reproducing kernel Hilbert spaces,” *IEEE Trans. Signal Process.*, 2015, to appear (doi: 10.1109/TSP.2015.2463261).