ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Online Learning with Operator-valued Kernels

Julien Audiffren[1] and Hachem Kadri[2] *

1- CMLA - ENS CACHAN
61 Avenue du president Wilson, 94230 Cachan - FRANCE
2- LIF - Aix Marseille Université
163 avenue de Luminy, 13288 Marseille - FRANCE

Abstract. We consider the problem of learning a vector-valued function f in an online learning setting. The function f is assumed to lie in a reproducing Hilbert space of operator-valued kernels. We describe an online algorithm for learning f while taking into account the output structure. This algorithm, OLOK, extends the standard kernel-based online learning algorithm NORMA from scalar-valued to operator-valued setting. We report a cumulative error bound that holds both for classification and regression. Our experiments show that the proposed algorithm achieves good performance results with low computational cost.

1 Introduction

We consider the problem of learning a function $f : \mathcal{X} \to \mathcal{Y}$ in a reproducing kernel Hilbert space, where $\mathcal{Y}$ is a Hilbert space with dimension $d > 1$. This problem has received relatively little attention in the machine learning community compared to the analogous scalar-valued case where $\mathcal{Y} \subseteq R$. In the last decade, more attention has been paid on learning vector-valued functions [1]. This attention is due largely to the developing of practical machine learning (ML) systems that can be suitably formulated as an optimization of vector-valued functions.

Motivated by the success of kernel methods in learning scalar-valued functions, in this paper we focus our attention to vector-valued function learning using reproducing kernels [2]. It is important to point out that in this context the kernel function outputs an operator rather than a scalar as usual. The operator allows to encode prior information about the outputs and their dependency. In contrast to scalar-valued kernels, operator-valued kernels provide a powerful way to extract relevant knowledge and encode the output structure when dealing with "complex" output learning problems. They have been applied with success in many learning contexts such as multi-task learning [3], functional response regression [4] and structured output prediction [5, 6].

Despite these recent advances, one major limitation with using operator-valued kernels is the high computational expense. Indeed, in contrast to the scalar-valued case, the kernel matrix associated to a reproducing operator-valued kernel is a block matrix of dimension $td \times td$, where t is the number of examples and d the dimension of the output space. Manipulating and inverting matrices of this size becomes particularly problematic when dealing with large t and d.

*Work partially supported by the CNRS PEPS project FLAME and by French ANR Projects LAMPADA (ANR-09-EMER-007) and GRETA (ANR-12-BS02-0004).

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

In this spirit we have asked whether, by learning the vector-valued function f in an online setting, one could develop efficient operator-valued kernel based algorithms with modest memory requirements and low computational cost.

In this context, it is worth mentioning that an online multi-task learning approach using a Bayesian framework and Gaussian processes was proposed in [7] and is related to our work. Compared to [7], our main contributions are: A) we propose an algorithm called OLOK, which extends the widely known NORMA algorithm [8] to operator-valued kernel setting (Section 2). OLOK does not require to invert the block kernel matrix associated to the operator-valued kernel and has at most a linear complexity with the number of examples at each update. B) We show theoretical bounds for OLOK (Section 2), and C) we provide an empirical evaluation of its performance which demonstrates its scalability and effectiveness on multi-output data sets (Section 3).

2 OLOK

Notation. Let $\mathcal{X}$ be a Polish space, $\mathcal{Y}$ a separable Hilbert space, and $\mathcal{H}$ a separable Reproducing Kernel Hilbert Space (RKHS) $\subset \mathcal{Y}^{\mathcal{X}}$ with $K : \mathcal{X} \times \mathcal{X} \to L(\mathcal{Y})$ its positive-definite reproducing operator-valued kernel. $L(\mathcal{Y})$ is the space of continuous endomorphisms of $\mathcal{Y}$ equipped with the operator norm. See [1] for more details on operator-valued kernels and their associated RKHSs. Let $t \in \mathbb{N}$ denotes the number of examples, $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ the i-th example, $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ a loss function, ∇ the gradient operator, and $R_{inst}(f, x, y) = \ell(f(x), y) + (\lambda/2)\|f\|^2_{\mathcal{H}}$ the instantaneous regularized error. $\lambda \in \mathbb{R}^+$ denotes the regularization parameter and $\eta_t \in \mathbb{R}^+$ the learning rate at time t, with $\eta_t \lambda < 1$.

As with scalar-valued kernels [8], the key idea here is to perform a stochastic gradient descent with respect to R_{inst}. The update rule (i.e., the definition of f_{t+1} as a function of f_t and the input (x_t, y_t)) is

$$f_{t+1} = f_t - \eta_{t+1} \nabla_f R_{inst}(f, x_{t+1}, y_{t+1})|_{f=f_t},$$

where η_{t+1} denotes the learning rate at time $t+1$. Using the reproducing property of positive-definite operator-valued kernels [1], it is easy to see that $\nabla_f(\ell(f(x), y)|_{f=f_t} = K(x, \cdot)\nabla_z \ell(z, y)|_{z=f_t(x)}$, from which we deduce

$$f_{t+1} = (1 - \eta_{t+1}\lambda)\, f_t - \eta_t K(x_{t+1}, \cdot)\nabla_z \ell(z, y)|_{z=f_t(x_{t+1})}. \quad (1)$$

As a consequence, if we choose $f_0 = 0$, then there exists $(\alpha_{i,j})_{i,j}$ a family of elements of $\mathcal{Y}$ such that $\forall t \geq 0$, $f_t = \sum_{i=1}^{t} K(x_i, \cdot)\alpha_{i,t}$. This leads to Algorithm 1, which we call OLOK (Online Learning with Operator-valued Kernels).

Truncation. OLOK algorithm described above needs to keep in memory all the previous input $\{x_i\}_{i=1}^t$ to compute the prediction value y_t, and this can be costly. However, the influence of these inputs decreases geometrically at each iteration, since $0 < 1 - \lambda\eta_t < 1$. Hence, the error induced by neglecting old terms can be controlled. So we can add a truncation step to store only a few relevant past observations (see the optional truncation step in Algorithm 1).

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Algorithm 1 OLOK

Input: $\lambda, \eta_t \in \mathbb{R}_+^*$, $(s_t)_t \in \mathbb{N}$, loss function ℓ,

Initialization: f_0=0

At time t: Input : (x_t, y_t)

1. **New coefficient:** $\alpha_{t,t} := -\eta_t \nabla_z \ell(z, y_t)|_{z=f_{t-1}(x_t)}$
2. **Update old coefficients:** $\forall 1 \leq i \leq t-1$, $\alpha_{i,t} := (1-\eta_t\lambda)\alpha_{i,t-1}$
3. **(Optional) Truncation:** $\forall 1 \leq i \leq t-s_t$, $\alpha_{i,t} := 0$
4. **Obtain** f_t**:** $f_t = \sum_{i=1}^{i=t} K(x_i, \cdot)\alpha_{i,t}$

Complexity analysis. We consider a naive implementation of OLOK when $\dim \mathcal{Y}(= d) < \infty$. At iteration t, the calculation of the prediction has complexity $O(td^2)$ and the update of the old coefficients has complexity $O(td)$. Hence the complexity up to iteration t is $O(t^2d^2)$. Note that the complexity of a naive implementation of the batch algorithm for learning with operator-valued kernels is $O(t^3d^3)$. A major advantage of OLOK is its lower computational complexity compared to classical batch operator-valued kernel-based algorithms.

Cumulative Error Bound. The cumulative error of the sequence $(f_i)_{i\leq t}$ given by OLOK is defined by $\sum_{i=1}^{t} \ell(f_i(x_i), y_i)$. It is interesting to compare this quantity to the error made with the function $\mathbf{g}_t$ obtained from a regularized empirical risk minimization algorithm which is the batch counterpart of OLOK. The function $\mathbf{g}_t$ is computed by solving the following minimization problem: $\mathbf{g}_t = \operatorname{argmin}_{h\in\mathcal{H}} R_{reg}(h,t) = \arg\min_{h\in\mathcal{H}} \frac{1}{t}\sum_{i=1}^{t} \ell(h(x_i),y_i) + \frac{\lambda}{2}\|h\|_{\mathcal{H}}^2$.

We analyze the cumulative error under the following common assumptions on the boundedness of the kernel and the C-admissibility of the loss.

Assumption 1 $\sup_{x\in\mathcal{X}} |K(x,x)|_{op} \leq \kappa^2$.

Assumption 2 *ℓ is C-admissible, i.e., ℓ is convex and C-Lipschitz with regard to its second variable.*

Theorem 1 *Let $\eta > 0$ such that $\eta\lambda < 1$. If Assumption 1 and Assumption 2 hold, then there exists $U > 0$ such that, with $\eta_t = \eta t^{-1/2}$,*

$$\frac{1}{t}\sum_{i=1}^{t} R_{inst}(f_i, x_i, y_i) \leq \inf_{g\in\mathcal{H}} R_{reg}(g,t) + \frac{\alpha}{\sqrt{t}} + \frac{\beta}{t},$$

where $\alpha = 2\lambda U^2(2\eta\lambda + 1/(\eta\lambda))$ (resp. $\alpha = 2\lambda U^2(10\eta\lambda + 1/(\eta\lambda))$ with the truncation step, and $s_t = \max\left(t^{1/2+\varepsilon}, t_0(\lambda,\eta,\varepsilon)\right)$ with $t_0(\lambda,\eta,\varepsilon) = \min\{t \in \mathbb{N}, \eta\lambda < \sqrt{t}, \exp(-\eta\lambda t^{\varepsilon}) \leq \eta\lambda t^{-0.5}, t^{0.5+\varepsilon} \leq 0.75t\}$), and $\beta = U^2/(2\eta)$.

The proof of Theorem 1 differs from that in the case of scalar-valued kernels [8] through several points which are grouped in Proposition 2.1. Due to the lack of space, we present here only the proof of Proposition 2.1 and refer the reader to [8] for more details.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Proposition 2.1 *If Assumption 1 and Assumption 2 hold, we have*

i. $\forall t \in \mathbb{N}^*,\ \|\alpha_{t,t}\|_{\mathcal{Y}} \leq \eta_t C$,

ii. *if* $\|f_0\|_{\mathcal{H}} \leq U = C\kappa/\lambda$ *then* $\forall t \in \mathbb{N}^*,\ \|f_t\|_{\mathcal{H}} \leq U$,

iii. $\forall t \in \mathbb{N}^*, \|\mathbf{g}_t\|_{\mathcal{H}} \leq U$.

PROOF: (i) is a direct consequence of the Lipschitz property of ℓ and the definition of α. (ii) is proved using Eq. (1) and (i) by induction on t. (ii) is true for $t = 0$. If (ii) is true for $t = m$, then it is true for $m + 1$, since $\|f_{m+1}\|_{\mathcal{H}} = \|\,(1 - \eta_t\lambda)\, f_m + k(x, \cdot)\alpha_{m,m}\|_{\mathcal{H}} \leq \frac{C\kappa}{\lambda} - \kappa\eta_t C + \kappa\eta_t C = \frac{C\kappa}{\lambda}$.
To prove (iii), one can see that by definition of $\mathbf{g}_t$, we have $\forall \varepsilon > 0$,

$$0 \leq \frac{\lambda}{2}(\|(1-\varepsilon)\mathbf{g}_t\|_{\mathcal{H}}^2 - \|\mathbf{g}_t\|_{\mathcal{H}}^2) + \frac{1}{t}\sum_{i=1}^{t} \ell((1-\varepsilon)\mathbf{g}_t(x_i), y_i) - \ell(\mathbf{g}_t(x_i), y_i)$$
$$\leq \frac{\lambda}{2}(\varepsilon^2 - 2\varepsilon)\|\mathbf{g}_t\|_{\mathcal{H}}^2 + C\varepsilon\kappa\|\mathbf{g}_t\|_{\mathcal{H}}.$$

Since this quantity must be positive for any $\varepsilon > 0$, the dominant term in the limit when $\varepsilon \to 0$ (i.e., the coefficient of ε) must be positive. Hence $\lambda\|\mathbf{g}_t\|_{\mathcal{H}} \leq C\kappa$. □

Case of the least squares loss. The least squares loss function does not satisfy Assumption 2. We provide here Assumption 3 which is a sufficient condition to recover the cumulative error bound in this case.

Assumption 3 $\ell(z, y) = \frac{1}{2}\|y - z\|_{\mathcal{Y}}^2$, $\exists C_y > 0$ *such that* $\forall t \geq 0$, $\|y_t\|_{\mathcal{Y}} \leq C_y$ *(the output is bounded), and* $\lambda > 2\kappa^2$.

Proposition 2.2 *If Assumption 1 and Assumption 3 hold, and* $\|f_0\|_{\mathcal{H}} < C_y/\kappa \leq U = \max(C_y/\kappa, 2C_y/\lambda)$, *then* $\forall t \in \mathbb{N}^*$*:*

i. $\|f_t\|_{\mathcal{H}} < C_y/\kappa \leq U$,

ii. $\exists V$ *in a 'neighbourhood' of* f_t *such that* $\ell(\cdot, y_t)|_V$ *is* $2C_y$ *Lipschitz,*

iii. $\|\alpha_{t,t}\|_{\mathcal{Y}} \leq 2\eta_t C_y$,

iv. $\|\mathbf{g}_t\|_{\mathcal{H}} \leq \dfrac{2C_y}{\lambda} \leq U$.

PROOF: We first prove that $\forall t \in \mathbb{N}^*$, $(i) \implies (ii) \implies (iii)$.
$(i) \implies (ii)$**:** in the least squares case, the application $z \mapsto \nabla_z \ell(z, y_t) = (z - y_t)$ is continuous. Assumptions 3 and 1 combined with (i) imply that $\|\nabla_z \ell(z, y_t)|_{z=f_t(x_t)}\|_{\mathcal{Y}} < 2C_y$. Using the continuity property, we obtain (ii).
$(ii) \implies (iii)$**:** using the same idea as in Proposition 2.1, we obtain $\|\alpha_{t,t}\|_{\mathcal{Y}} = \|\eta_t \nabla_z \ell(z, y_t)|_{z=f_{t-1}(x_t)}\|_{\mathcal{Y}} \leq \eta_t 2C_y$. Now we prove (i) by induction. The initialization $(t = 0)$ is a consequence of the hypothesis, $\|f_0\|_{\mathcal{H}} < C_y/\kappa$. For the propagation $(t = m)$: If $\|f_m\|_{\mathcal{H}} < C_y/\kappa$, then using (ii) and (iii), we obtain $\|f_{m+1}\|_{\mathcal{H}} \leq (1 - \eta_t\lambda)\, C_y/\kappa + 2\kappa\eta_t C_y = C_y/\kappa + \eta_t C_y(2\kappa - \lambda/\kappa) < C_y/\kappa$, where the last transition is a consequence of Assumption 3. Finally, note that by definition of $\mathbf{g}_t$, since $0 \in \mathcal{H}$, $\frac{\lambda}{2}\|\mathbf{g}_t\|_{\mathcal{H}}^2 + \frac{1}{t}\sum_{i=1}^{t} \ell(\mathbf{g}_t(x_i), y_i) \leq \frac{\lambda}{2}\|0\|_{\mathcal{H}}^2 + \frac{1}{t}\sum_{i=1}^{t} \ell(0, y_i) \leq C_y^2$.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

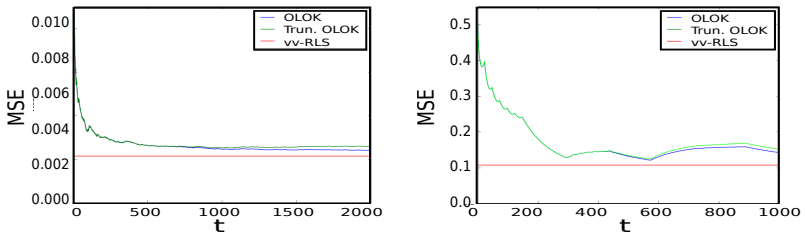

Fig. 1: Variation of the MSE of OLOK with the number of examples. (left) Synthetic data set. (right) Parkinson Telemonitoring Data set.

3 Experiments

In this section, we conduct experiments on synthetic and real-world datasets to evaluate the efficiency of OLOK. We compare the original and the truncated version of OLOK with its batch counterpart. We consider here the vector-valued Regularized Least Squares algorithm (vv-RLS). To measure the performance of the learning algorithms, we use the Mean Square Error (MSE).

We use the Gaussian kernel with parameter μ for OLOK and vv-RLS defined by $K_\mu(x, x') = \exp(-\|x - x'\|_2^2/\mu)\mathbf{J}$, where μ is the parameter of the kernel varying from 10^{-3} to 10^2 and $\mathbf{J}$ denotes the $d \times d$ matrix with coefficient $\mathbf{J}_{i,j}$ equals to 1 if $i = j$ and 1/10 otherwise. Compared to the scalar-valued case, the kernel here outputs a matrix which allows to take into account output dependencies. The regularization parameter λ for vv-RLS is chosen using five-fold cross validation. For OLOK, we use $\eta = 0.1$, $\eta_t = 1/\sqrt{t}$, and $\lambda = 3$.

We run OLOK and vv-RLS algorithms on the following real-world data sets: Wine Quality[2] (4898 instances, 12 attributes, $d=2$), Parkinson Telemonitoring (5875 instances, 20 attributes, $d=2$), and Advertisement click rate[3] (1 millions instance, 120 attributes, $d=16$). Additionally, we also used a synthetic dataset (10000 instances, 50 attributes, $d=20$) generated as described in [9].

Convergence. We depict in Figure 1 the evolution of the MSE of OLOK as the number of training data available increases, as well as the MSE of vv-RLS. We can see that the MSE of OLOK quickly reaches the value of its batch counterpart. The MSE of truncated OLOK is not far from that of vv-RLS and also "decreases" when more samples are available. The convergence performance of truncated OLOK is however less good than that of OLOK.

Accuracy and performance. We report the averaged MSE and the STD (standard deviation) of each algorithm and their respective running time in Table 1. These results show that OLOK achieves a good trade-off between speed and performance. It performs nearly as good as vv-RLS while being much faster. Truncated OLOK allows to improve the speed of OLOK to the detriment of the accuracy, but the MSE difference between the two algorithms is relatively small.

[2]http://archive.ics.uci.edu/ml/datasets.
[3]https://www.kaggle.com.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

Table 1: MSE (± STD) and running time (RT) for vv-RLS, OLOK and Truncated OLOK (T-OLOK).

	Synthetic		Wine		Parkinson		Click	
	MSE	RT	MSE	RT	MSE	RT	MSE	RT
vv-RLS	1e-5±1e-6	550s	2e-4±2e-5	240s	9.1e-2±1e-4	120s	-	-
OLOK	1e-5±1e-6	17s	2.3e-4±2e-5	12s	9.5e-2±1e-3	20s	1e-4	5h
T-OLOK	5e-5±4e-6	4s	2.3e-4±2e-5	2s	1.0e-1±1e-3	6s	3e-4	30m

4 Conclusion

The main barrier to wider use of operator-valued kernels is their computational demands. In this paper we have addressed this issue using an online learning framework. We have presented a new algorithm OLOK that compares favorably to its batch counterpart in terms of running time while having a similar accuracy performance. As future work, we plan to extend the framework of multiple operator-valued kernel learning [10] to the online setting.

References

[1] C. A. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17:177–204, 2005.

[2] M. A. Álvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: a review. *Foundation and Trends in Machine Learning*, 4(3):195–266, 2012.

[3] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

[4] H. Kadri, E. Duflos, P. Preux, S. Canu, and M. Davy. Nonlinear functional regression: a functional RKHS approach. AISTATS, 2010.

[5] C. Brouard, F. d'Alche Buc, and M. Szafranski. Semi-supervised penalized output kernel regression for link prediction. ICML, 2011.

[6] H. Kadri, M. Ghavamzadeh, and P. Preux. A generalized kernel approach to structured output learning. ICML, 2013.

[7] G. Pillonetto, F. Dinuzzo, and G. De Nicolao. Bayesian online multitask learning of gaussian processes. *PAMI, IEEE Transactions on*, 32(2):193–205, 2010.

[8] J. Kivinen, A. J. Smola, and R. C. Williamson. Online learning with kernels. *Signal Processing, IEEE Transactions on*, 52(8):2165–2176, 2004.

[9] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[10] H. Kadri, A. Rakotomamonjy, F. Bach, and P. Preux. Multiple operator-valued kernel learning. NIPS, 2012.