ONLINE LEARNING WITH KERNELS
A NEW APPROACH FOR SPARSITY CONTROL BASED ON A COHERENCE CRITERION

Jean-Baptiste Pothin, Cédric Richard

Institut Charles Delaunay (ICD-M2S, FRE CNRS 2848)
Université de Technologie de Troyes, 12 rue Marie Curie, BP 2060, 10010 Troyes cedex - France
jean_baptiste.pothin@utt.fr cedric.richard@utt.fr

ABSTRACT

Kernel methods are well known standard tools for solving function approximation and pattern classification problems. In this paper, we consider online learning in a reproducing kernel Hilbert space. We develop a simple and computationally efficient algorithm for sparse solutions. The approach is based on sequential projection learning and the coherence criterion, which is a fundamental parameter to characterize dictionaries of functions in sparse approximation problems. Experimental results show the effectiveness of our approach.

1. INTRODUCTION

Kernel methods have been successfully applied to a large class of problems; see [1] for a recent survey. The attractiveness of such algorithms stem from their elegant treatment of nonlinear problems and their connection with statistical learning theory [2]. However, a notable limitation of kernel methods is their computational complexity since the amount of computer memory and training time typically increase superlinearly with the number of observations. By noting that this challenge is closely related to the *sparsity* of the solution, several authors have proposed learning algorithms including sparsity control mechanisms [3, 4, 5].

Recently a theoretical foundation for online function estimation in reproducing kernel Hilbert spaces (RKHS) was proposed [6], leading to a highly efficient method known as sequential projection learning (SPL). This approach is based on stochastic gradient descent (SGD) and orthogonal projections. Kernel basis functions that do not contribute significantly to the performance of the model are discarded to produce a sparse solution, via incremental and decremental steps. This strategy is similar to that employed in the sparse online Gaussian process framework described in [7]. It is also related to the kernel recursive least-squares (KRLS) algorithm [8], although no decremental step is required here. Experimental results demonstrate that SPL performs well on synthetic and real data [9]. However, the decremental step is particularly computational expensive since it requires as many matrix inversions as there are kernel basis functions in the model. In this paper, we propose an alternative online function estimation strategy that differs from SPL by the novelty condition used to assess the impact of kernel basis functions on the performance of the model. It is based on the coherence criterion, which was shown to be a fundamental parameter to characterize dictionaries of functions in sparse approximation problems, see [10] for a complete description. It was introduced as a quantity of heuristic interest for Matching Pursuit in [11]. The first theoretical developments were described in [12], and enriched for Basis Pursuit in [13], [14].

The rest of this paper is organized as follows. In Section 2, we briefly review sequential projection learning in RKHS. Our sparsity control mechanism based on the coherence criterion is presented in Section 3. Its effectiveness is confirmed through simulations in Section 4.

2. SEQUENTIAL PROJECTION LEARNING

In this paper, we consider sparse online learning with kernels. The goal is to approximate a mapping $f^* : \mathcal{X} \rightarrow \mathbb{R}$ based on a sequence of input-output pairs $(\boldsymbol{x}_t, y_t) \in \mathcal{X} \times \mathbb{R}$ that become available one by one. The output of the learning algorithm is commonly called *hypothesis* and the set of all possible hypotheses is denoted by $\mathcal{H}$. Assuming that $\mathcal{H}$ is a RKHS means that there exists a kernel function $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ and a dot product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ such that (i) κ has the reproducing property $\langle f, \kappa(\boldsymbol{x}, \cdot) \rangle_{\mathcal{H}} = f(\boldsymbol{x})$, $\forall \boldsymbol{x} \in \mathcal{X}$, (ii) $\mathcal{H}$ is the closure of the span of the $\kappa(\boldsymbol{x}, \cdot)$'s. In this context, any function $f \in \mathcal{H}$ can be expressed as a linear combinations of kernel functions [15]. Starting from an initial arbitrary hypothesis $f_0 \in \mathcal{H}$, it is desired that the learning algorithm produces a sequence $f_1, ..., f_t$ where f_t is the hypothesis learnt from data received up to time t, namely,

$$f_t(\cdot) = \sum_{i=1}^{m} \alpha_i \kappa(\tilde{\boldsymbol{x}}_i, \cdot), \tag{1}$$

where $\alpha_i \in \mathbb{R}$ and $\tilde{\boldsymbol{x}}_i \in \mathcal{X}$. Note the difference in notation between the samples $\tilde{\boldsymbol{x}}_i$, ordered as they are inserted into the expansion, and the samples $\boldsymbol{x}_t$ available at each time t. Also note that the model order is m, and not t, as we will subsequently introduce a sparsity control mechanism.

2.1. Stochastic Gradient Descent in RKHS

A natural measure of quality for f_t is the *instantaneous risk* defined by

$$g_{t+1}(f_t) \triangleq \frac{1}{2}(f_t(\boldsymbol{x}_{t+1}) - y_{t+1})^2, \tag{2}$$

that is, the squared error between the model output f_t at time instant $t+1$ and the desired output. The SGD update rule is given by

$$f_{t+1} = f_t - \eta_t \nabla_f g_{t+1}(f_t), \tag{3}$$

where $\eta_t > 0$ is the learning rate and ∇_f is the gradient with respect to f. We have

$$f_{t+1} = f_t - \eta_t (f_t(\boldsymbol{x}_{t+1}) - y_{t+1})\kappa(\boldsymbol{x}_{t+1}, \cdot). \tag{4}$$

In the stationary case, it has been shown that $\|f^* - f_t\| \to 0$ provided that the following simple condition on the learning rate η_t is satisfied [16]:

$$0 < \eta_t < \frac{2}{\kappa(\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+1})} \quad \forall t. \tag{5}$$

Starting from $f_0 = 0$, i.e., $\boldsymbol{\alpha}_0 = \varnothing$, the update rule (4) can be summarized as

$$\boldsymbol{\alpha}_{t+1} \leftarrow \begin{pmatrix} \boldsymbol{\alpha}_t \\ \eta_t e_t \end{pmatrix} \tag{6}$$

with $e_t = y_{t+1} - f_t(\boldsymbol{x}_{t+1})$, and $\tilde{\boldsymbol{x}}_{t+1} = \boldsymbol{x}_{t+1}$. The computational complexity of this naive algorithm then grows as more data points become available, which is a significant problem for online applications.

2.2. Sparse projection learning

To avoid inserting a kernel function into the expansion (1) at each time instant, sparsification methods based on novelty criteria have been proposed. For instance, Dodd *et al.* extend the model f_t with $\kappa(\boldsymbol{x}_{t+1}, \cdot)$ if, and only if, [6]

$$\min_{\boldsymbol{\beta}} \|f_{t+1} - \sum_{i=1}^{m} \beta_i \kappa(\tilde{\boldsymbol{x}}_i, \cdot)\|_{\mathcal{H}} > \epsilon_0, \tag{7}$$

where ϵ_0 is a positive threshold determining the sparsity level. Let $\boldsymbol{K}_{a,b}$ be the a-by-b Gram matrix $\boldsymbol{K}_{a,b}(i,j) = \kappa(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j)$ with $1 \leq i \leq a$ and $1 \leq i \leq b$. Provided that $\boldsymbol{K}_{m,m}^{-1}$ is invertible, it can be shown that [6]

$$\boldsymbol{\beta} = \boldsymbol{K}_{m,m}^{-1} \boldsymbol{K}_{m,m+1} \begin{pmatrix} \boldsymbol{\alpha}_t \\ \eta_t e_t \end{pmatrix}. \tag{8}$$

If condition (7) is satisfied, f_{t+1} is updated according to the rule (6). Otherwise, f_{t+1} is obtained as follows, $\boldsymbol{\alpha}_{t+1} \leftarrow \boldsymbol{\beta}$, without additional computational effort. This is known as the *incremental* step. Upon adding new kernels, there is the possibility for existing kernels to become redundant. The basic *decremental* step determines kernel basis functions $\kappa(\tilde{\boldsymbol{x}}_i, \cdot)$ which do not contribute significantly to the performance of the model. This stage consists of removing each of the kernel basis functions in turn and comparing the reduced models to the initial one. This is the most expensive part of the algorithm since $\boldsymbol{\beta}$ must be calculated for each reduced model.

3. SPARSITY CONTROL USING COHERENCE

Coherence is a fundamental parameter to characterize dictionaries of functions in sparse approximation problems, see [10] for an extensive description. It is defined as the maximum absolute inner product between two unit-norm functions a given dictionary $\mathcal{D}_m$. It reflects the most extreme correlations in the dictionary and, consequently, it is equal to zero for every orthonormal basis. In our kernel-based context, dictionary unit-norm functions are given by $\kappa(\tilde{\boldsymbol{x}}_i, \cdot)/\|\kappa(\tilde{\boldsymbol{x}}_i, \cdot)\|$ and the coherence parameter is defined as

$$\mu = \max_{i \neq j} \left| \left\langle \frac{\kappa(\tilde{\boldsymbol{x}}_i, \cdot)}{\|\kappa(\tilde{\boldsymbol{x}}_i, \cdot)\|}, \frac{\kappa(\tilde{\boldsymbol{x}}_j, \cdot)}{\|\kappa(\tilde{\boldsymbol{x}}_j, \cdot)\|} \right\rangle_{\mathcal{H}} \right| = \max_{i \neq j} |\rho_{ij}|, \tag{9}$$

where $\rho_{ij} = \kappa(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j)/\sqrt{\kappa(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_i)\kappa(\tilde{\boldsymbol{x}}_j, \tilde{\boldsymbol{x}}_j)}$. Note that μ can be easily determined from the Gram matrix $\boldsymbol{K}_{m,m}$. In particular, in the case of a unit-norm kernel κ, it is the largest absolute value of the off-diagonal entries of $\boldsymbol{K}_{m,m}$. Without loss of generality, we will assume in what follows that κ is a unit-norm kernel in order to simplify expressions.

3.1. Incremental step

Let φ_{t+1} be the kernel function $\kappa(\boldsymbol{x}_{t+1}, \cdot)$. We suggest inserting it into $\mathcal{D}_m = \{\tilde{\varphi}_1, \ldots, \tilde{\varphi}_m\}$ provided that the coherence of $\mathcal{D}_{m+1}$ remains below a threshold $\mu_0 \in [0, 1[$, namely,

$$\max_i |\langle \tilde{\varphi}_i, \varphi_{t+1} \rangle_{\mathcal{H}}| = \|\boldsymbol{\rho}_{t+1}\|_\infty < \mu_0, \tag{10}$$

where $\boldsymbol{\rho}_{t+1}$ is the column vector of dimension m whose i^{th} component is $\rho_{i,t+1} = |\langle \tilde{\varphi}_i, \varphi_{t+1} \rangle_{\mathcal{H}}|$. The parameter μ_0 determines both the level of sparsity and the maximum coherence of $\mathcal{D}_m$. We have established that this condition guarantees the finiteness of the dictionary. In addition, under a mild technical condition, we have derived an analytic relationship between ϵ_0 in (7) and μ_0 in (10). Due to lack of space, these results will be presented in a companion paper. Consider the case when condition (10) does not hold for φ_{t+1}. From (8), it follows that the best approximation of f_{t+1} onto the span of f_t is parameterized by [6]

$$\boldsymbol{\alpha}_{t+1} = \boldsymbol{K}_{m,m}^{-1} [\boldsymbol{K}_{m,m} \ \boldsymbol{\rho}_{t+1}] \begin{pmatrix} \boldsymbol{\alpha}_t \\ \eta_t e_t \end{pmatrix} \tag{11}$$

$$= \boldsymbol{\alpha}_t + \eta_t e_t \boldsymbol{\nu}, \tag{12}$$

with $\boldsymbol{\nu} = \boldsymbol{K}_{m,m}^{-1} \boldsymbol{\rho}_{t+1}$. Consider now the case when condition (10) holds for φ_{t+1}. The latter is then incorporated into the dictionary, namely, $\mathcal{D}_{m+1} = \mathcal{D}_m \cup \{\varphi_{t+1}\}$, and $\boldsymbol{\alpha}_{t+1}$ is

updated according to the rule (6). As for basic SPL, the matrix inversion process can be performed efficiently by use of a rank one update. Let $\boldsymbol{K}_{m+1,m+1}$ be the Gram matrix of the dictionary $\mathcal{D}_{m+1}$. We have

$$\boldsymbol{K}_{m+1,m+1} = \begin{pmatrix} \boldsymbol{K}_{m,m} & \boldsymbol{\rho}_{t+1} \\ \boldsymbol{\rho}_{t+1}^T & 1 \end{pmatrix}. \tag{13}$$

The inverse of $\boldsymbol{K}_{m+1,m+1}$ can be computed as follows

$$\boldsymbol{K}_{m+1,m+1}^{-1} = \frac{1}{\lambda}\begin{pmatrix} \lambda \boldsymbol{K}_{m,m}^{-1} + \boldsymbol{\nu}\boldsymbol{\nu}^T & -\boldsymbol{\nu} \\ -\boldsymbol{\nu}^T & 1 \end{pmatrix}, \tag{14}$$

where $\lambda = 1 - \boldsymbol{\rho}_{t+1}^T \boldsymbol{\nu}$. Note that the novelty condition (10) is an $O(m)$ operation. If φ_{t+1} is retained, the main computational effort is the rank one update, i.e, $O(m^2)$. Otherwise, it is the projection (12), which is $O(m^2)$. Therefore, the proposed incremental step is an $O(m^2)$ operation.

3.2. Decremental step

A common strategy which ensures that the model order m is bounded is to discard a kernel function from the expansion whenever m exceeds a predefined threshold m_0. Here we suggest to discard the kernel function $\tilde{\varphi}_{i_0}$ which leads the coherence of the dictionary to decrease, that is,[1]

$$i_0 = \arg\max_{i, i \neq j} |\rho_{ij}|. \tag{15}$$

Once $\tilde{\varphi}_{i_0}$ has been removed from $\mathcal{D}_m$, the inverse of the matrix $\boldsymbol{K}_{m-1,m-1}$ must be calculated in order to update the model f_t. Let us introduce the following notations

$$\boldsymbol{K}_{m,m} = \begin{pmatrix} \boldsymbol{K}_{m-1,m-1} & \boldsymbol{\rho}_{i_0} \\ \boldsymbol{\rho}_{i_0}^T & 1 \end{pmatrix}, \tag{16}$$

$$\boldsymbol{K}_{m,m}^{-1} = \begin{pmatrix} \boldsymbol{Q}_{m-1,m-1} & \boldsymbol{q}_0 \\ \boldsymbol{q}_0^T & q_{i_0} \end{pmatrix}, \tag{17}$$

and

$$\boldsymbol{\alpha}_t = \begin{pmatrix} \boldsymbol{\alpha}_{t\setminus\{i_0\}} \\ \alpha_{i_0} \end{pmatrix}, \tag{18}$$

where the initial i_0^{th} column and row (resp. element) of the matrices $\boldsymbol{K}_{m,m}$ and $\boldsymbol{K}_{m,m}^{-1}$ (resp. the vector $\boldsymbol{\alpha}_t$) are placed in the last position. From the decomposition method [7], it follows that

$$\boldsymbol{K}_{m-1,m-1}^{-1} = \boldsymbol{Q}_{m-1,m-1} - \frac{\boldsymbol{q}_0 \boldsymbol{q}_0^T}{q_{i_0}}. \tag{19}$$

Finally, a similar calculation to (12) shows that the reduced order model f_{t+1} is parameterized by

$$\boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_{t\setminus\{i_0\}} + \alpha_{i_0} \boldsymbol{K}_{m-1,m-1}^{-1} \boldsymbol{\rho}_{i_0}. \tag{20}$$

Finding the index i_0 of the kernel function to be discarded, updating the inverse matrix and calculating $\boldsymbol{\alpha}_{t+1}$ are procedures of computational complexity $O(m^2)$. After a transient period, the computational effort per time-step, including the incremental stage, is thus $O(m_0^2)$.

[1] Note that $j_0 = \arg\max_{j, i \neq j} |\rho_{ij}|$ could also be considered.

4. EXPERIMENTS

We consider first as a benchmark problem the nonlinear time series described by the following difference equation

$$\begin{aligned} y_t = &(0.8 - 0.5\exp(-y_{t-1}^2))y_{t-1} \\ &- (0.3 + 0.9\exp(-y_{t-1}^2))y_{t-2} + 0.1\sin(\pi y_{t-1}). \end{aligned}$$

The kernel function was chosen to be of the form

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2) \tag{21}$$

where $\boldsymbol{x}_i = (y_{i-1}, y_{i-2})^T$. We then generated 300 data points from the initial point $(0.1, 0.1)$. The first 200 data points were used as a training set while the last 100 data points were used to estimate the prediction error :

$$\text{NRMSE} = \frac{1}{\sigma^2 M} \sum_{i=1}^{M} (\hat{y}_{t+i} - y_{t+i})^2, \tag{22}$$

where M is the prediction horizon ($M = 100$), σ^2 is the variance of the true data and $\hat{y}_{t+i} = f_t(\boldsymbol{x}_{t+i})$ is the predicted output made by the hypothesis learnt from the training data. We first applied basic SPL including incremental and decremental steps based on novelty condition (7). The hyperparameters of the algorithm were fixed as in [6], i.e., $\gamma = 3.73$ and $\epsilon_0 = 0.01$. This resulted in sparse solution involving 24 kernels out of the possible 200 and the NRMSE was found to be equal to $7.07 \cdot 10^{-4}$. Note that this is significantly better than [6] where NRMSE $= 6.1 \cdot 10^{-3}$ with 47 kernels retained[2]. We then applied SPL with a sparsity control mechanism based on our μ_0-coherent approach with $m_0 = 24$ and $\mu_0 = 0.75$. In that case, we obtained the smallest prediction NRMSE $= 6.02 \cdot 10^{-4}$. In order to assess the performance of our algorithm in a noisy case, the data were corrupted with additive gaussian noise $\mathcal{N}(0, 0.01)$ and each algorithms were parameterized as above. On the one hand, standard SPL led to NRMSE $= 0.057$ with 42 terms retained. On the other hand, SPL with our sparsity control mechanism gave NRMSE $= 0.0598$. This is larger than basic SPL. However, note that the order of the kernel expansion provided by basic SPL was nearly two more times larger.

As a second benchmark, we consider the nonlinear time series described by the following difference equation

$$y_t = 0.5y_{t-1} + 0.3y_{t-1}u_t + 0.2u_t + 0.05y_{t-1}^2 + 0.6u_t^2,$$

with initial condition $y_1 = 10$. The observations were generated as $\boldsymbol{x}_t = y_t + \epsilon_t$ with u_t and ϵ_t i.i.d zero-mean Gaussian random variables with variances 0.1 and 0.05, respectively. The Gaussian kernel (21) was used with $\gamma = 5$. The threshold μ_0 used to assess the novelty of the basis function in (10) was made adaptive by setting $\mu_0 = \mu_t$, where μ_t is the coherence

[2] In [6], SPL is considered without decremental step.

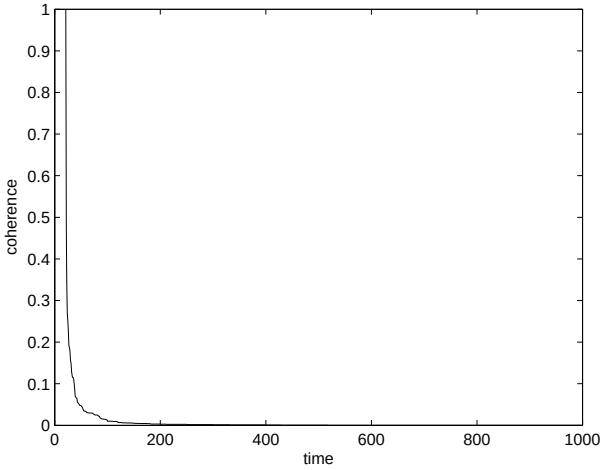

Fig. 1. Mean evolution of the coherence of the dictionary.

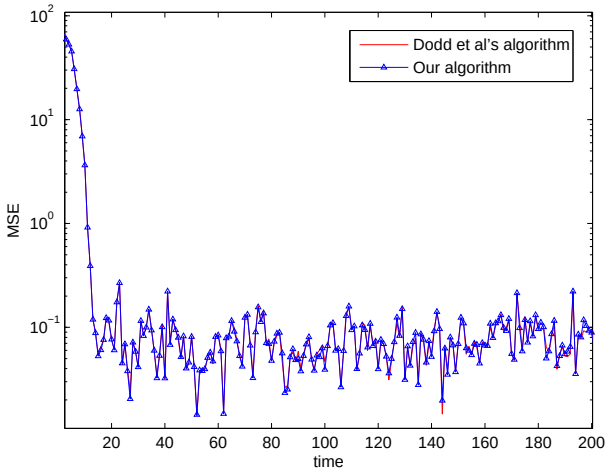

Fig. 2. Predicted output error as a function of time.

of the dictionary at the t^{th} iteration. This ensures the quasi-incoherence of the dictionary over time, i.e., as more samples become available, previous basis functions are replaced by nearby orthogonal basis functions. Figure 1 shows the mean evolution of μ_t over 10 simulations with $m_0 = 20$. We note that the dictionary quickly becomes quasi-incoherent. In particular, μ_t was found to be about $1.5 \cdot 10^{-4}$ after 200 iterations. Figure 2 shows that both basic SPL and μ_0-SPL have quite the same convergence behavior. However, the computational cost of our approach is $O(m^2)$ whereas the complexity of basic SPL is $O(m^3)$.

5. REFERENCES

[1] J. P. Vert, K. Tsuda, and B. Schölkopf, "A primer on kernel methods," in *Kernel Methods in Computational Biology*, B. Schölkopf, K. Tsuda, and J. P. Vert, Eds. Cambridge, MA: The MIT Press, 2004, pp. 35–70.

[2] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY: Springer, 1995.

[3] P. B. Nair, A. Choudhury, and A. J. Keane, "Some greedy learning algorithms for sparse regression and classification with mercer kernels," *Journal of Machine Learning Research*, vol. 3, pp. 781–801, 2002.

[4] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 2165–2176, 2004.

[5] K. Crammer, J. Kandola, and Y. Singer, "Online classification on a budget," *Advances in Neural Information Processing Systems*, vol. 16, 2004.

[6] T. J. Dodd, V. Kadirkamanathan, and R. F. Harrison, "Function estimation in hilbert space using sequential projections," 2003, pp. 113–118.

[7] L. Csato and M. Oper, "Sparse representation for gaussian process models," *Advances in Neural Information Processing Systems 13*, pp. 444–450, 2001.

[8] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2275–2285, 2004.

[9] T. J. Dodd, B. Mitchinson, and R. F. Harrison, "Sequential kernel methods: a multiple model approach to hyperparameters," *Proceedings of The Second International Conference on Computational Intelligence*, 2003.

[10] J. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[11] S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[12] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.

[13] M. Elad and A. M. Bruckstein, "A generalized uncertainty principle and sparse representations in pairs of bases," *IEEE Transactions on Information Theory*, vol. 48, no. 9, pp. 2558–2567, 2002.

[14] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization," in *Proc. National Academy of Sciences of the USA*, vol. 3, 2003, pp. 2197–2202.

[15] B. Scholkopf and A. J. Smola, "Learning with kernels." Cambridge, MA: The MIT Press, 2002.

[16] T. Dodd, S. Nair, R. Harrison, V. Kadirkamanathan, and S. Phonphitakchai, "Sparse, on-line learning in reproducing kernel hilbert spaces," *IEEE Transactions on Signal Processing*, submitted in 2005.