2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Bayesian online multi-task learning using regularization networks

Gianluigi Pillonetto, Francesco Dinuzzo and Giuseppe De Nicolao

Abstract— **Recently, standard single-task kernel methods have been extended to the case of multi-task learning under the framework of regularization. Experimental results have shown that such an approach can perform much better than single-task techniques, especially when few examples per task are available. However, a possible drawback may be computational complexity. For instance, when using regularization networks, complexity scales as the cube of the overall number of data associated with all the tasks. In this paper, an efficient computational scheme is derived for a widely applied class of multi-task kernels. More precisely, a quadratic loss is assumed and the multi-task kernel is the sum of a common term and a task-specific one. The proposed algorithm performs on-line learning recursively updating the estimates as new data become available. The learning problem is formulated in a Bayesian setting. The optimal estimates are obtained by solving a sequence of subproblems which involve projection of random variables onto suitable subspaces. The algorithm is tested on a simulated data set.**

Index Terms— **multi-task learning; machine learning; kernel methods; regularization; Bayesian estimation; Kalman filtering**

I. Introduction

The usual regression learning problem has to do with reconstructing a multi-dimensional real-valued function from discrete and noisy samples [1]. An interesting extension is the so-called multi-task learning problem in which several multi-dimensional functions (tasks) are simultaneously estimated. For the problem to be significant it is necessary to assume that the tasks are related to each other in some way so that measurements taken on a task are informative with respect to the other ones.

Important examples of multi-task learning are encountered in biomedicine when multiple experiments are performed in subjects from a population [2]. In fact, similar patterns are observed in individual responses so that data from a subject can help reconstructing also the responses of other individuals. In pharmacokinetics (PK) and pharmacodynamics (PD) the joint analysis of several individual curves is currently used and goes under the name of population analysis [3]. In this field, the adopted models are parametric, e.g. compartmental ones, so that data depend nonlinearly on the parameters [4], [5]. The development of the NONMEM software traces back to the seventies [6], [7] whereas more sophisticated approaches include also Bayesian MCMC algorithms [8], [9]. Only recently, machine learning/nonparametric approaches have been proposed for the population analysis of PK/PD data [10], [11], [12].

In the machine learning literature, the term multi-task learning was originally introduced in [13]. A series of works have pointed out the performance improvement achievable by using a multi-task approach instead of a single-task one which learns the functions separately [14], [15]. A Bayesian treatment was developed in [16] where bounds are obtained on the amount of information needed to learn a task when it is simultaneously learned with several other tasks. More recently, in [17] a regularized kernel method has been proposed that relies on the theory of vector-valued Reproducing kernel Hilbert spaces [18].

Among the open research topics formulated in [17] there are computational complexity and development of on-line multi-task learning schemes. A drawback of proposed multi-task learning schemes is in fact the number of operations required to achieve the estimates that may be much larger than that involved by independent learning of the single tasks. For instance, when using regularization networks, complexity scales with the cube of the overall number of examples. In [17] it is stated that, when all the k tasks share the same n inputs and the multi-task kernel has a suitable structure, the complexity can be reduced to $O(kn^3)$. In fact, an $O(kn^3)$ algorithm for regularization networks in the longitudinal case can be found in [19]. *On-line* multi-task learning occurs when a set of examples for a new task is made available in real-time. There is an obvious interest for the development of effective (recursive) learning algorithms also for this kind of problem.

The aforementioned open issues are addressed in this paper for a widely applied class of multi-task problems, characterized by quadratic loss and kernels which are the sum of a common term and a task-specific one. In particular, we develop a computationally efficient algorithm that solves the on-line multi-task learning problem. The proposed algorithm recursively updates the estimates as new data become available either for a new or an existing task. No constraints are posed on the location of the input samples. The algorithm exploits a Bayesian reformulation of the problem and the estimates are recursively updated by solving a sequence of subproblems involving projections of random variables onto suitable subspaces. A key technical lemma developed in [12] is used to compute some of the projections. Remarkably enough, part of the overall scheme can be interpreted as a Kalman filter operating on a system whose dimension grows over time. Efficient formulas for the computation of confidence intervals are also worked out.

G. Pillonetto (giapi@dei.unipd.it) is with Dipartimento di Ingegneria dell'Informazione, University of Padova, Padova, Italy.

F. Dinuzzo (francesco.dinuzzo@gmail.com) and G. De Nicolao (giuseppe.denicolao@unipv.it) are with Dipartimento di Informatica e Sistemistica, University of Pavia, Pavia, Italy.

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

The paper is so organized. In Section 2, a brief review on the nonlinear multi-task learning problem is presented. In Section 3, the multi-task learning problem is stated within a Bayesian framework. In Section 4, some notation used in the paper is introduced and some technical results useful for development of the new computational scheme are derived. In Section 5 and 6, an efficient algorithm which solves the on-line multi-task learning problem is worked out, while in Section 7 simulated data are used to test the computational scheme. Conclusions then end the paper.

II. A brief review of kernel-based multi-task learning

We are given a set of k task functions $\mathbf{f}_i : X \mapsto \Re$ where X denotes a compact set in $\Re^d$. Notice that it is assumed that there is a common input space for all the tasks. We assume that for the i-th task the following n_i examples are available

$$(x_{1,i}, y_{1,i}), (x_{2,i}, y_{2,i}), ..., (x_{n_i,i}, y_{n_i,i}), \tag{1}$$

Our aim is to jointly estimate all the unknown functions $\mathbf{f}_i$ starting from these examples. To do this, we start by following the approach described in [17] where the vector-valued function $\mathbf{f} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_k]$ belongs to a Reproducing Kernel Hilbert Space (RKHS) of vector-valued functions defined on X. Letting W denote a suitable Hilbert space with inner product $< .,. >_W$, it can be shown that there exist operators $\Theta_i : X \mapsto W, i = 1, ..., k$ such that

$$\mathbf{f}_i(x) =< w, \Theta_i(x) >_W \qquad x \in X \quad i = 1, 2, ..., k$$

In addition, W is isometrically isomorphic to an RKHS whose (multi-task) reproducing kernel is given by

$$K((x,l),(s,q)) =< \Theta_l(x), \Theta_q(s) >_W \qquad x, s \in X$$

where $l = 1, ..., k$ and $q = 1, ..., k$. According to the regularization approach, the vector $w \in W$ can be obtained by minimizing the single-task functional

$$J(w) = \sum_{i=1}^{k} \sum_{l=1}^{n_i} V(y_{l,i}, \mathbf{f}_i(x)) + \gamma \|w\|^2_W$$

In the above equation, $V : \Re^2 \mapsto \Re$ denotes a loss function which penalizes solutions which are not able to well account for the experimental evidence, while γ is the regularization parameter which has to balance the training error and the solution smoothness measured by $\|w\|^2_W$. Under rather general conditions on V, the representer theorem can be exploited to obtain the following expression for the minimizer of J (see e.g. [20]):

$$\tilde{\mathbf{f}}_j(x) = \sum_{i=1}^{k} \sum_{l=1}^{n_i} c_{l,i} K\left((x,j),(x_{l,i},i)\right) \quad x \in X, j = 1, ..., k \tag{2}$$

where $\{c_{l,i}\}$ are suitable scalars. In particular, if square errors are used, i.e. $V(a,b) = (a-b)^2$ for every $a, b \in \Re$, the solution is given by a regularization network whose weights $\{c_{l,i}\}$ are the solution of the following linear system of equations

$$\sum_{i=1}^{k} \sum_{l=1}^{n_i} \left[K((x_{l,i},i),(x_{j,q},q)) + \gamma \delta_{lj} \delta_{iq}\right] c_{l,i} = y_{j,q} \tag{3}$$

where $q = 1, ..., k$, $j = 1, ..., n_q$ and δ_{ij} is the Kroenecker delta.

III. Problem formulation in a Bayesian setting

In the sequel, we use $E[.]$ to denote the expectation operator and vectors are column vectors, unless otherwise specified. In addition, given two random vectors q and w, we define $cov[q,w] = E[(q - E[q])(w - E[w])^T]$ and $Var[q] = E[(q - E[q])(q - E[q])^T]$. We also use I_n to denote the identity matrix of size $n \times n$.

From now, we assume that the following relation holds

$$y_{j,i} = \mathbf{f}_i(x_{j,i}) + \epsilon_{j,i} \tag{4}$$

where $\{\epsilon_{j,i}\}$ is white Gaussian noise of variance σ^2.

Definition 1: We define the following vectors

$$\begin{aligned} y_i &= [y_{1,i} \ldots y_{n_i,i}]^T \qquad y^k = [y_1^T \ldots y_k^T]^T \\ \epsilon_i &= [\epsilon_{1,i} \ldots \epsilon_{n_i,i}]^T \qquad x_i = [x_{1,i} \ldots x_{n_i,i}]^T \end{aligned}$$

while x^k corresponds to the vector whose components are given by the elements (with no repetitions) of the set $\bigcup_{i=1}^{k} x_i$.

■

Notice that the cardinality of y^k, denoted by n_{y^k}, is $\sum_{i=1}^{k} n_i$, while the cardinality of x^k, denoted as n_{x^k}, can be much smaller than n_{y^k}.

The following proposition relies upon the duality between Gaussian processes and RKHS, e.g. see [21]. It provides a link between the solution of a regularization network associated with a multi-task kernel and Bayesian estimation of Gaussian random fields.

Proposition 2: Assume that $\{\mathbf{f}_i\}$ are zero-mean Gaussian random fields with covariances defined by

$$cov\left(\mathbf{f}_i(x), \mathbf{f}_q(s)\right) = K((x,i),(s,q)) \qquad x, s \in X$$

where $i = 1, ..., k$ and $q = 1, ..., k$. Let also (4) hold where $\{\epsilon_{j,i}\}$ are independent of $\{\mathbf{f}_i\}$. Then, for $j = 1, ..., k$, the minimum variance estimate of $\mathbf{f}_j$ conditioned on y^k is defined by (2,3) once γ is set to σ^2.

■

The following assumption introduces the specific class of multi-task kernels which will be the focus of the paper.

Assumption 3: For each i and $x \in X$, we have

$$\mathbf{f}_i(x) = \bar{\mathbf{f}}(x) + \hat{\mathbf{f}}_i(x)$$

where $\bar{\mathbf{f}}$ and $\{\hat{\mathbf{f}}_i\}$ are zero-mean Gaussian random fields. We also assume that $\{\epsilon_{j,i}\}$, $\bar{\mathbf{f}}$ and $\{\hat{\mathbf{f}}_i\}$ are all mutually independent.

■

Assumption 3 extends the model described in Section 3.1.1 in [17] to nonlinear multi-task kernels (in a stochastic setting). In particular, a kernel is defined for each task which is a convex combination of two kernels. The first one, if used alone, would correspond to learning independent tasks and, in our Bayesian framework, is associated with the auto-covariance of $\hat{\mathbf{f}}_i$. The second one, if used alone, would correspond to assuming that all tasks are actually the same and is defined by the auto-covariance of $\bar{\mathbf{f}}$. The use of the convex combination of these two kernels amounts to assuming that each task is given by the sum of an average function and an individual shift specific for each task, see e.g. [12].

When examples from k tasks are available and Proposition 2 is exploited, it would seem that the computational complexity for obtaining the optimal function estimates scales as the cube of n_{y^k} which is the cost of solving (3). The rest of the paper is devoted to derive a more efficient numerical scheme that exploits the specific structure of the problem coming from Assumption 3. In addition, the goal is to perform estimation in an online manner, as formalized below.

Problem 4: Fix k. For arbitrary $x \in X$ and integer j, compute efficiently $E[\mathbf{f}_j(x)|y^k]$. Further, suppose that a new set of examples relative to task $k+1$ becomes available. Then, compute efficiently $E[\mathbf{f}_j(x)|y^{k+1}]$.

IV. Preliminary results

We start by providing some new notation that will be used in the sequel.

Definition 5: Let $\bar{f}^k$ be the vector whose components are the elements of the set $\{\bar{\mathbf{f}}(x), x \in x^k\}$. The components of the vectors $\bar{f}_k$ and $\hat{f}_k$ are instead defined by the sets $\{\bar{\mathbf{f}}(x), x \in x_k\}$ and $\{\hat{\mathbf{f}}_k(x), x \in x_k\}$, respectively.

■

It comes from the definition above that

$$y_k = \bar{f}_k + \hat{f}_k + \epsilon_k \tag{5}$$

Notice also that, for a suitable matrix F^k and random vector $\hat{\epsilon}^k$, with $\hat{\epsilon}^k$ independent of $\bar{\mathbf{f}}$, we can write

$$y^k = F^k \bar{f}^k + \hat{\epsilon}^k \qquad \hat{\epsilon}^k \perp \bar{\mathbf{f}} \tag{6}$$

The following three lemmas will prove useful in the following.

Lemma 6: We have

a) $E\left[\bar{\mathbf{f}}(x)|\bar{f}^k, y^k\right] = E\left[\bar{\mathbf{f}}(x)|\bar{f}^k\right]$ $\forall x \in X$, and in particular $E\left[\bar{f}^{k+1}|\bar{f}^k, y^k\right] = E\left[\bar{f}^{k+1}|\bar{f}^k\right]$

b) $E\left[\hat{\mathbf{f}}_j(x)|\bar{f}_j, y^k\right] = E\left[\hat{\mathbf{f}}_j(x)|\bar{f}_j, y_j\right]$

■

Lemma 7: We have

$$\begin{aligned} Var\left[y_{k+1}|y^k\right] &= Var\left[\bar{f}_{k+1}|y^k\right] + Var\left[\hat{f}_{k+1}\right] \\ &+ Var\left[\epsilon_{k+1}\right] \\ cov\left[\bar{f}^{k+1}, y_{k+1}|y^k\right] &= cov\left[\bar{f}^{k+1}, \bar{f}_{k+1}|y^k\right] \\ E\left[y_{k+1}|y^k\right] &= E\left[\bar{f}_{k+1}|y^k\right] \end{aligned}$$

Proof: It suffices to exploit (5), replacing y_{k+1} with $\bar{f}_{k+1} + \hat{f}_{k+1} + \epsilon_{k+1}$, and recall that the latter three terms are zero-mean and mutually independent. ■

In the equations below, we use $\mathbf{N}(\mu, \Sigma)$ to denote the multinormal density with mean μ and covariance Σ. The following lemma is an immediate extension of that reported in Appendix of [12]. We just stress that, differently from the statement in [12], the symbol z here denotes a vector (in place of a scalar) and the weaker condition $V > 0$ (in place of $\Sigma > 0$) is invoked. The proof remains however identical.

Lemma 8: Let y, v and η be random vectors and F be a matrix such that

$$y = F\eta + v \qquad v \sim \mathbf{N}(0, \Sigma_v) \qquad \Sigma_v > 0$$

Let also

$$\begin{bmatrix} z \\ \eta \end{bmatrix} \sim \mathbf{N}(0, \Sigma) \qquad \Sigma = \begin{bmatrix} U & \Gamma \\ \Gamma^T & V \end{bmatrix} \qquad v \perp \begin{bmatrix} z \\ \eta \end{bmatrix}$$

$$Var[z] = U \qquad Var[\eta] = V \qquad V > 0$$

Then

$$Var[z|y] = Var[z|\eta] + Var[E[z|\eta]|y]$$

where

$$\begin{aligned} Var[z|\eta] &= U - \Gamma V^{-1} \Gamma^T \\ Var[E[z|\eta]|y] &= \Gamma V^{-1} Var[\eta|y] V^{-1} \Gamma^T \end{aligned}$$

V. Recursive computation of the posterior mean and autocovariance of $\bar{f}^k$

In this section we derive the recursive update formula for $E\left[\bar{f}^k|y^k\right]$ and $Var\left[\bar{f}^k|y^k\right]$, as the number k of tasks and corresponding examples increase. As it will be clear in the sequel, these are the two key quantities to be propagated over time in order to compute efficiently $E\left[\mathbf{f}_j(x)|y^k\right]$ (for arbitrary j and $x \in X$). In particular, our numerical scheme consists of the three steps listed below.

1) Initialization. This amounts to determine $E\left[\bar{f}^1|y_1\right]$ and $Var\left[\bar{f}^1|y_1\right]$
2) Update related to possibly new inputs locations. This consists of extending $E\left[\bar{f}^k|y^k\right]$ and $Var\left[\bar{f}^k|y^k\right]$ to x^{k+1}, i.e. for known $E\left[\bar{f}^k|y^k\right]$ and $Var\left[\bar{f}^k|y^k\right]$ one has to compute $E\left[\bar{f}^{k+1}|y^k\right]$ and $Var\left[\bar{f}^{k+1}|y^k\right]$
3) Measurement update. This consists of determining $E\left[\bar{f}^{k+1}|y^{k+1}\right]$ and $Var\left[\bar{f}^{k+1}|y^{k+1}\right]$ as a function of $E\left[\bar{f}^{k+1}|y^k\right]$ and $Var\left[\bar{f}^{k+1}|y^k\right]$

A. Initialization

The following proposition solves Step 1 above.

Proposition 9: Let

$$A = Var\left[\bar{f}_1\right] + Var\left[\hat{f}_1\right] + Var\left[\epsilon_1\right]$$

Then, we have

$$\begin{aligned} E\left[\bar{f}_1|y_1\right] &= Var\left[\bar{f}_1\right] A^{-1} y_1 \\ Var\left[\bar{f}_1|y_1\right] &= Var\left[\bar{f}_1\right] - Var\left[\bar{f}_1\right] A^{-1} Var\left[\bar{f}_1\right] \end{aligned}$$

Proof: Exploiting well known results on estimation of joint Gaussian vectors, see e.g. [22], [23], one has

$$\begin{aligned} E\left[\bar{f}_1|y_1\right] &= cov\left[\bar{f}_1, y_1\right] \left(Var\left[y_1\right]\right)^{-1} y_1 \\ Var\left[\bar{f}_1|y_1\right] &= Var\left[\bar{f}_1\right] - cov\left[\bar{f}_1, y_1\right] \left(Var\left[y_1\right]\right)^{-1} \\ &\times \; cov\left[\bar{f}_1, y_1\right]^T \end{aligned}$$

Using the equation $y_1 = \bar{f}_1 + \hat{f}_1 + \epsilon_1$ and the independence assumptions, one immediately obtains

$$\begin{aligned} cov\left[\bar{f}_1, y_1\right] &= Var\left[\bar{f}_1\right] \\ Var\left[y_1\right] &= Var\left[\bar{f}_1\right] + Var\left[\hat{f}_1\right] + Var\left[\epsilon_1\right] \end{aligned}$$

which completes the proof. ■

B. Update for handling new input locations

The following proposition provides the solution of Step 2. It is worth stressing that the numerical procedure described below is different from the predictor step in a Kalman filter since the dimension of the state (i.e. the number of distinct input locations up to the first k tasks) can increase. This nontrivial issue will be handled by means of the projection Lemma 8.

Proposition 10: Let ζ be a vector such that $\bar{f}^{k+1} = [\zeta^T \quad (\bar{f}^k)^T]^T$ and define

$$H_k = \left[\begin{array}{c} cov\left[\zeta, \bar{f}^k\right] Var\left[\bar{f}^k\right]^{-1} \\ I_{n^k} \end{array} \right]$$

Then, we have

$$\begin{aligned} E\left[\bar{f}^{k+1}|y^k\right] &= H_k E\left[\bar{f}^k|y^k\right] \quad (7) \\ Var\left[\bar{f}^{k+1}|y^k\right] &= Var\left[\bar{f}^{k+1}\right] - H_k cov\left[\bar{f}^{k+1}, \bar{f}^k\right]^T \\ &+ H_k Var\left[\bar{f}^k|y^k\right] H_k^T \quad (8) \end{aligned}$$

Proof: To derive (7), we first project $\bar{f}^{k+1}$ first onto the space generated by $\bar{f}^k$ and y^k and then onto y^k, i.e. we write

$$E\left[\bar{f}^{k+1}|y^k\right] = E\left[E\left[\bar{f}^{k+1}|\bar{f}^k, y^k\right]|y^k\right]$$

Using Lemma 6 (point a), we obtain

$$\begin{aligned} E\left[\bar{f}^{k+1}|\bar{f}^k, y^k\right] &= E\left[\bar{f}^{k+1}|\bar{f}^k\right] \\ &= cov\left[\bar{f}^{k+1}, \bar{f}^k\right] \left(Var\left[\bar{f}^k\right]\right)^{-1} \bar{f}^k \end{aligned}$$

Projecting the result onto y^k and observing that

$$H_k = cov\left[\bar{f}^{k+1}, \bar{f}^k\right] \left(Var\left[\bar{f}^k\right]\right)^{-1}$$

we also obtain

$$E\left[E\left[\bar{f}^{k+1}|\bar{f}^k, y^k\right]|y^k\right] = E\left[H_k \bar{f}^k|y^k\right] = H_k E\left[\bar{f}^k|y^k\right]$$

which proves (7).
To obtain (8), recall from (6) that $y^k = F^k \bar{f}^k + \hat{\epsilon}^k$ with $\hat{\epsilon}^k \perp \bar{f}^{k+1}$. Then, (8) is obtained from Lemma 8, with the following assignments

$$\begin{aligned} z &= \bar{f}^{k+1} \qquad \eta = \bar{f}^k \qquad v = \hat{\epsilon}^k \\ U &= Var\left[\bar{f}^{k+1}\right] \quad V = Var\left[\bar{f}^k\right] \quad \Gamma = cov\left[\bar{f}^{k+1}, \bar{f}^k\right] \end{aligned}$$

■

C. Measurement update

The following result whose proof is omitted for reasons of space performs the measurement update required by Step 3.

Proposition 11: Let

$$A_k = Var\left[\bar{f}_k|y^{k-1}\right] + Var\left[\hat{f}_k\right] + Var\left[\epsilon_k\right]$$

Then, we have

$$\begin{aligned} E\left[\bar{f}^{k+1}|y^{k+1}\right] &= E\left[\bar{f}^{k+1}|y^k\right] + cov\left[\bar{f}^{k+1}, \bar{f}_{k+1}|y^k\right] \\ &\times \; A_{k+1}^{-1}\left(y_{k+1} - E\left[\bar{f}_{k+1}|y^k\right]\right) \quad (9) \end{aligned}$$

$$\begin{aligned} Var\left[\bar{f}^{k+1}|y^{k+1}\right] &= Var\left[\bar{f}^{k+1}|y^k\right] \\ &- cov\left[\bar{f}^{k+1}, \bar{f}_{k+1}|y^k\right] A_{k+1}^{-1} cov\left[\bar{f}^{k+1}, \bar{f}_{k+1}|y^k\right]^T \quad (10) \end{aligned}$$

VI. Solution of the online multi-task learning problem

The following proposition is the main result of the paper. It shows that $E\left[\mathbf{f}_j(x)|y^k\right]$ admits a representation in terms of a regularization network whose $(n_{x^k} + n_j)$-dimensional weight vector can be efficiently updated online as the number of tasks and associated examples increase over time. In particular, given k tasks, the complexity of the proposed algorithm is $O(kn_{x^k}^3)$. Recall that the number of distinct input locations n_{x^k} may well be much smaller than the overall number of examples n_{y^k}.

Proposition 12: Let

$$\bar{f}^k = \left[\bar{f}_1^k, \bar{f}_2^k, ..., \bar{f}_{n_{x^k}}^k\right]^T \qquad \hat{f}_j = \left[\hat{f}_{1,j}, \hat{f}_{2,j}, ..., \hat{f}_{n_j,j}\right]^T$$

Then, it holds that

$$E\left[\mathbf{f}_j(x)|y^k\right] = \sum_{i=1}^{n_{x^k}} a_i cov\left[\bar{\mathbf{f}}(x), \bar{f}_i^k\right] + \sum_{i=1}^{n_j} b_i cov\left[\hat{\mathbf{f}}(x), \hat{f}_{i,j}\right]$$

where, letting

$$a = [a_1, ..., a_{n_{x^k}}]^T \qquad b = [b_1, ..., b_{n_j}]^T$$

the weights of the regularization network are defined by

$$\begin{aligned} a &= \left(Var\left[\bar{f}^k\right]\right)^{-1} E\left[\bar{f}^k|y^k\right] \\ b &= \left(Var\left[\hat{f}_j\right] + Var\left[\epsilon_j\right]\right)^{-1} \left(y_j - E\left[\bar{f}_j|y^k\right]\right) \end{aligned}$$

and can be efficiently updated by means of Propositions 10 and 11.

Proof: We have

$$E\left[\mathbf{f}_j(x)|y^k\right] = E\left[\overline{\mathbf{f}}(x)|y^k\right] + E\left[\hat{\mathbf{f}}_j(x)|y^k\right] \qquad (11)$$

As far as the first term on the right of (11) is concerned, following the same arguments used in the first part of the proof of Proposition 10, one obtains

$$E\left[\overline{\mathbf{f}}(x)|y^k\right] = cov\left[\overline{\mathbf{f}}(x), \bar{f}^k\right]\left(Var\left[\bar{f}^k\right]\right)^{-1} E\left[\bar{f}^k|y^k\right]$$

from which the expression for a comes immediately.
In order to compute $E\left[\hat{\mathbf{f}}_j(x)|y^k\right]$, we still project twice. In particular, we first project $\hat{\mathbf{f}}_j(x)$ onto the space spanned by $\bar{f}_j$ and y^k and then onto y^k, i.e. it holds that

$$E\left[\hat{\mathbf{f}}_j(x)|y^k\right] = E\left[E\left[\hat{\mathbf{f}}_j(x)|y^k, \bar{f}_j\right]|y^k\right]$$

Using Lemma 6 (point b) and recalling that $y_j = \bar{f}_j + \hat{f}_j + \epsilon_j$, one obtains

$$\begin{aligned} E\left[\hat{\mathbf{f}}_j(x)|y^k, \bar{f}_j\right] &= E\left[\hat{\mathbf{f}}_j(x)|y_j, \bar{f}_j\right] \\ &= cov\left[\hat{\mathbf{f}}_j(x), \hat{f}_j\right] \\ &\times \left(Var\left[\hat{f}_j\right] + Var\left[\epsilon_j\right]\right)^{-1}\left(y_j - \bar{f}_j\right) \end{aligned}$$

Projecting $\left(y_j - \bar{f}_j\right)$ onto y^k, we have

$$\begin{aligned} E\left[\hat{\mathbf{f}}_j(x)|y^k\right] &= cov\left[\hat{\mathbf{f}}_j(x), \hat{f}_j\right]\left(Var\left[\hat{f}_j\right] + Var\left[\epsilon_j\right]\right)^{-1} \\ &\times \left(y_j - E\left[\bar{f}_j|y^k\right]\right) \end{aligned}$$

which completes the proof. ■

A. Computation of confidence intervals

Assume that data y^k relative to the first k tasks have been already processed and that $Var\left[\bar{f}_l|y^k\right]$ for $l = 1, 2, ..., k$ has been obtained by using Propositions 10 and 11. Now, we consider the problem of computing the posterior variance $Var\left[\mathbf{f}_l(x)|y^k\right]$ for any $x \in X$ and integer l, so as to obtain confidence intervals for $\mathbf{f}_l(x)$.
To this aim, it is useful to define

$$\bar{\tau}_l = \begin{bmatrix} \bar{f}_l \\ \overline{\mathbf{f}}_l(x) \end{bmatrix} \quad \hat{\tau}_l = \begin{bmatrix} \hat{f}_l \\ \hat{\mathbf{f}}_l(x) \end{bmatrix} \quad \tau_l = \bar{\tau}_l + \hat{\tau}_l$$

Let also P be a matrix such that

$$y_l = P\bar{\tau}_l + \hat{f}_l + \epsilon_l = P\tau_l + \epsilon_l$$

Further, compute $Var\left[\bar{\tau}_l|y^k\right]$ by means of Lemma 8. The solution of the problem is then provided by the following proposition, whose proof is omitted for reasons of space.

Proposition 13: Define M_l as

$$\left(\left(Var\left[\bar{\tau}_l|y^k\right]\right)^{-1} - P^T\left(Var\left[\hat{f}_l\right] + Var\left[\epsilon_l\right]\right)^{-1} P\right)^{-1} + Var\left[\hat{\tau}_l\right]$$

Then, it holds that

$$Var\left[\tau_l|y^k\right] = M_l - M_l P^T\left(PM_lP^T + Var\left[\epsilon_l\right]\right)^{-1} PM_l$$

VII. Numerical example

We consider the problem of learning online 20 mono-dimensional tasks. Let $X = [0, 10]$ and let also the auto-covariances of $\overline{\mathbf{f}}$ and $\hat{\mathbf{f}}_i$ be given by Gaussian kernels. To be specific

$$\begin{aligned} cov\left[\overline{\mathbf{f}}(q), \overline{\mathbf{f}}(w)\right]\right) &= e^{-(q-w)^2} \\ cov\left[\hat{\mathbf{f}}_i(q), \hat{\mathbf{f}}_i(w)\right]\right) &= e^{-\left(\frac{q-w}{0.4}\right)^2} \qquad i = 1, 2, ..., 20 \end{aligned}$$

We assume that the vectors x_i of input locations contain a common subset of 100 input locations which are distributed uniformly on X. Other input values are randomly chosen and then added to the inputs locations so as to obtain $n_i = 120$ for $i \in \{1, 4, 6, 9, 11, 14, 16, 19\}$, $n_i = 110$ for $i \in \{2, 5, 7, 10, 12, 15, 17, 20\}$ and $n_i = 140$ for $i \in \{3, 8, 13, 18\}$. Estimates of the task functions have to be computed starting from measurements corrupted by a white Gaussian noise with a constant standard deviation equal to 0.2. Notice that in this case $n_{y^{10}} = 2400$ while $n_{x^{20}} = 500$. Thus, the standard algorithm would obtain the regularization network weights by inverting a large matrix of size 2400×2400, while using our method the solution can be obtained by inverting matrices whose dimensions are always less than 500×500.
In Fig. 1, we display the true function $\overline{\mathbf{f}}$ (thick line) and its estimate $E\left[\overline{\mathbf{f}}|y^k\right]$ (solid line), together with 95% confidence intervals (dashed lines), for increasing values of k. It is seen that the estimator exploits data relative to new incoming tasks in order to improve the quality of the average curve estimate and reduce the uncertainty around it. Finally, in the top and bottom panels of Fig. 2, we also display the true task functions $\mathbf{f}_3$ and $\mathbf{f}_5$ (thick line), their noisy samples (circles), and the corresponding optimal estimates (solid line).

VIII. Conclusions

Often, it is not possible to collect a large number of examples in order to learn accurately a function. When several related functions have to be learned simultaneously the use of multi-task learning in place of standard single-task methods may be of great value. However, the computational complexity may significantly increase. For example, for regularized kernel methods with square loss functions (i.e. regularization networks), the number of operations scales with the cube of the overall number of examples.
Recent work in the literature has proposed effective learning

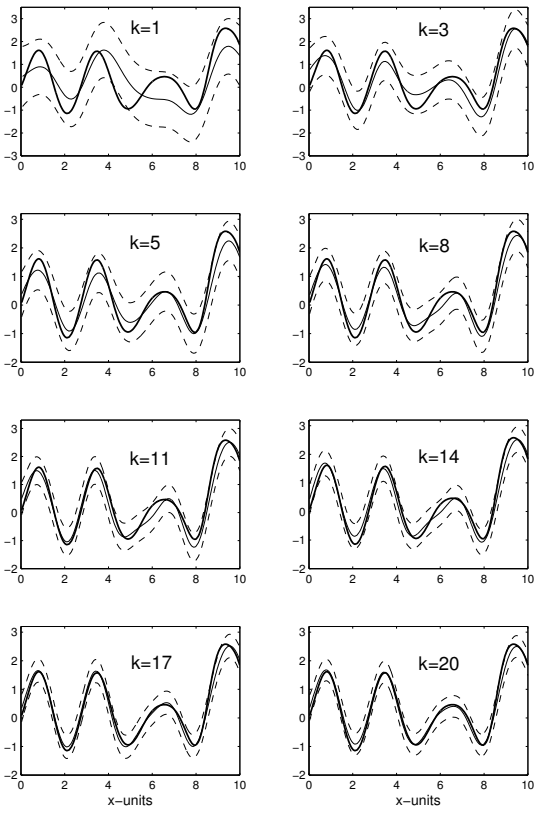

Fig. 1. True average $\bar{\mathbf{f}}$ (thick line) and optimal estimates for increasing values of k (thin line) with 95% confidence intervals (dashed lines)

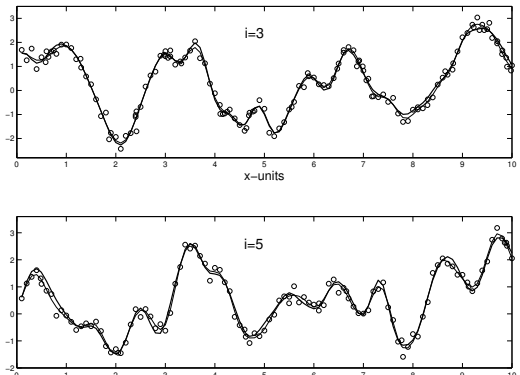

Fig. 2. True $\mathbf{f}_i$ (thick line), noisy samples (circles) and optimal estimates (thin line) for task $\#3$ (top panel) and task $\#5$ (bottom panel)

methods when the multi-task kernel is a suitable combination of two kernels [12], [19]. In this paper, a new efficient algorithm has been developed in a Bayesian setting. In particular, the proposed scheme, exploiting the existence of common input locations among the tasks, reduces the size of the matrices which have to be inverted to obtain the estimates without assuming as in [19] that all tasks share the same input locations. Our approach reconstructs the task functions by solving a sequence of subproblems which involve projections of random variables onto suitable subspaces spanned by the data. This leads to an incremental on-line algorithm which updates the regularization network weights as new task examples become available over time. Efficient formulas for the computation of confidence intervals have also been worked out.

ACKNOWLEDGMENT

This research has been partially supported by the European Commission under project HYGEIA (NEST-4995), by FIRB Project Learning theory and application, and by the PRIN Project New Methods and Algorithms for Identification and Adaptive Control of Technological Systems.

REFERENCES

[1] T. Poggio and F. Girosi. Networks for approximation and learning. In *Proc. IEEE*, volume 7, pages 1481–1497, 1990.

[2] F. Ferrazzi, P. Magni, and R. Bellazzi. Bayesian clustering of gene expression time series. In *Proc. of 3rd Int. Workshop on Bioinformatics for the Management, Analysis and Interpretation of Microarray Data (NETTAB 2003)*, pages 53–55, 2003.

[3] L. B. Sheiner. The population approach to pharmacokinetic data analysis: rationale and standard data analysis methods. *Drug Metabolism Reviews*, 15:153–171, 1994.

[4] M. Davidian and D. M. Giltinan. *Nonlinear Models for Repeated Measurement Data*. Chapman and Hall, New York, 1995.

[5] J.A. Jacquez. *Compartmental analysis in biology and medicine*. Ann Arbor: University of Michigan Press, 1985.

[6] L. B. Sheiner, B. Rosenberg, and V. V. Marathe. Estimation of population characteristics of pharmacokinetic parameters from routine clinical data. *J. Pharmacokin. Biopharm.*, 5(5):445–479, 1977.

[7] S. Beal and L. Sheiner. *NONMEM User's Guide*. NONMEM Project Group, University of California, San Francisco, 1992.

[8] J. C. Wakefield, A. F. M. Smith, A. Racine-Poon, and A. E. Gelfand. Bayesian analysis of linear and non-linear population models by using the Gibbs sampler. *Applied Statistics*, 41:201–221, 1994.

[9] D. J. Lunn, N. Best, A. Thomas, J. C. Wakefield, and D. Spiegelhalter. Bayesian analysis of population PK/PD models: general concepts and software. *J. Pharmacokinet. Pharmacodyn.*, 29(3):271–307, 2002.

[10] P. Magni, R. Bellazzi, G. De Nicolao, I. Poggesi, and M. Rocchetti. Nonparametric AUC estimation in population studies with incomplete sampling: a Bayesian approach. *J. Pharmacokin. Pharmacodyn.*, 29(5/6):445–471, 2002.

[11] M. Neve, G. De Nicolao, and L. Marchesi. Nonparametric identification of population models: An MCMC approach. *IEEE Trans. on Biomedical engineering*, 55:41–50, 2008.

[12] M. Neve, G. De Nicolao, and L. Marchesi. Nonparametric identification of population models via Gaussian processes. *Automatica*, 43(7):1134–1144, 2007.

[13] R. Caruana. Multi-task learning. *Machine Learning*, 28:41–75, 1997.

[14] S. Thrun and L. Pratt. *Learning to learn*. Kluwer, 1997.

[15] B. Bakker and T. Heskes. Task clustering and gating for bayesian multi-task learning. *Journal of Machine Learning Research*, (4):83–99, 2003.

[16] J. Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, (28):7–39, 1997.

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

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

[19] G. De Nicolao, G. Pillonetto, M. Chierici, and C. Cobelli. Efficient nonparametric population modeling for large data sets. In *Proc. of American Control Conference, 2007, New York, USA*, pages 2921–2926, 2007.

[20] B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *Proceedings of the Annual Conference on Computational Learning Theory*, pages 416–426, Portland, OR, USA, 2001.

[21] G. Wahba. *Spline Models for Observational Data*. SIAM, Philadelphia, 1990.

[22] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, N.J., USA, 1979.

[23] A. N. Shiryaev. *Probability*. Springer, New York, NY, USA, 1996.