Learning using state space kernel machines

Aleksandr Y. Aravkin ∗ Bradley M. Bell ∗∗ James V. Burke ∗∗∗
Gianluigi Pillonetto ****

Abstract: Reconstruction of a function from noisy data is often formulated as a regularized optimization problem whose solution closely matches an observed data set and also has a small reproducing kernel Hilbert space norm. The loss functions that measure agreement between the data and the function are often smooth (e.g. the least squares penalty), but non-smooth loss functions are of interest in many applications. Using the least squares penalty, large machine learning problems with kernels amenable to a stochastic state space representation (which we call state space kernel machines) have been solved using a Kalman smoothing approach. In this paper we extend this approach for state space kernel machines to the Vapnik penalty, a particularly important non-smooth penalty that is robust to outlier noise in the measurements and induces a sparse representation of the reconstructed function. We exploit the structure of such models using interior point methods that efficiently solve the functional recovery problem with the Vapnik penalty, and demonstrate the effectiveness of the method on a numerical experiment. The computational effort of our approach scales linearly with the number of data points.

Keywords: kernel based regularization; Kalman smoothing; Gaussian processes; statistical learning; estimation theory; regularization networks; support vector regression; robust estimators

1. INTRODUCTION

Minimizing a regularization functional in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ associated with a symmetric and positive-definite kernel $K$ has become a popular approach to reconstruct a scalar function from noisy data, e.g. see (Aronszajn, 1950; Schölkopf and Smola, 2001; Wahba, 1990; Girosi et al., 1995). In particular, regularization in $\mathcal{H}$ estimates the unknown function as

$$\hat{f}(\cdot) = \arg\min_{f(\cdot)} \sum_{i=1}^{N} \rho[z_i - f(t_i)] + \gamma \|f(\cdot)\|_{\mathcal{H}}^2,$$  

(1)

where $\gamma \in \mathbb{R}^+$ is the regularization parameter, $t_i$ is the location where the measurement $z_i$ is collected, $\rho(\cdot)$ is the so called loss function and $\|\cdot\|_{\mathcal{H}}$ is the RKHS norm induced by the reproducing kernel $K(\cdot,\cdot)$, see (Aronszajn, 1950). The solution of (1) can also be interpreted in terms of Bayesian estimation. Under suitable assumptions on the noise model, it represents the maximum a posteriori estimate of a zero-mean Gaussian random field of covariance proportional to $K(\cdot,\cdot)$ (Hengland, 2007). This connection is described in more detail in Section 2. One of the important features of the above approach is that, even if the dimension of $\mathcal{H}$ is infinite, the solution is a linear combination of a finite number of kernel functions centered at the points $\{t_i\}$. In fact, under mild assumptions on the loss, according to the representer theorem (Wahba, 1998; Schölkopf et al., 2001) solutions to (1) can be expressed as kernel sections centered on the sampling locations, i.e.

$$\hat{f}(\cdot) = \sum_{i=1}^{N} \tilde{c}_i K(t_i,\cdot),$$  

(2)

where $\tilde{c}_i$ are suitable scalars. For instance, (2) holds when $\rho(\cdot)$ is quadratic, defining a regularization network (Poggio and Girosi, 1990), or is the Vapnik’s $\varepsilon$-insensitive loss, leading to support vector regression (Vapnik, 1998; Evgeniou et al., 2000; Gunter and Zhu, 2006). In particular, the Vapnik penalty has recently been object of considerable attention, due to its sparsity inducing properties (many $\tilde{c}_i$ in (2) turn out to be zero) and its robustness with respect to outliers relative to the quadratic loss.

In many scientific fields such as telecommunication, biology and biomedicine, one is faced with the problem of reconstructing the (possibly multi-dimensional) input of a dynamic linear system from noisy output data (Bertero, 1989; De Vito et al., 2005). In addition, the system model is often formulated in state space (Pillonetto and Saccomani, 2006). Without loss of generality, in this paper we focus on single input single output systems, restricting our attention to monodimensional regression problems. We assume that the kernel $K(\cdot,\cdot)$ is given in state space form, see e.g. Section 4 in (De Nicolao and Ferrari Trecate, 2003), and for such problems we use kernel-based regularization to reconstruct $f(\cdot)$ using (1). More precisely, we model $f(\cdot)$ as a realization of a nonstationary scalar Gaussian process (sampled at times $\{t_i\}$) defined as follows:
\textbf{2. CONNECTION WITH BAYESIAN REGULARIZATION AND KALMAN SMOOTHING}

In this section, we describe how to use Kalman smoothing for general loss functions to solve the functional recovery problem (1). In our notation, vectors are column vectors. If \(a\) and \(b\) are random vectors, \(p(a)\) is the probability density for \(a\), \(E[a]\) is the expected value of \(a\),

\[
V(a, b) = E \left[ (a - E[a])(b - E[b])^T \right]
\]

is the covariance of \(a\) with \(b\), \(V(a) = V(a, a)\) is the variance of \(a\), and \(\max_y p(a | b)\) is the maximum (with respect to \(a\)) of the conditional probability density for a given \(b\). We begin with two lemmas which are instrumental in proving Proposition 2.4, the main result of this section. The proof of the first lemma relies upon well known properties of joint Gaussian vectors, see e.g. (Anderson and Moore, 1979), and is therefore omitted.

\textbf{Lemma 2.1.} Suppose that \(u\) and \(y\) are jointly Gaussian random vectors. It follows that the maximum of \(p(y | u)\) with respect to \(y\) does not depend on the value of \(u\) and is given by

\[
\max_{y} p(y | u) = \exp \left( - \frac{1}{2} \det \left\{ 2\pi V(y) - V(y, u) V(u)^{-1} V(u, y) \right\} \right).
\]

\textbf{Lemma 2.2.} Suppose that \(u\) and \(y\) are jointly Gaussian random vectors, and that \(z\) is a random vector for which \(p(z | u, y) = p(z | u)\). Given a realization of \(z\), define corresponding estimates for \(u\) and \(y\) by

\[
(\hat{u}, \hat{y}) = \arg\max_{u, y} p(z, u, y),
\]

where we assume that the solution \((\hat{u}, \hat{y})\) is unique. It follows that

\[
\hat{u} = \arg\max_{u} p(z | u) p(u) \quad \text{and} \quad \hat{y} = \arg\max_{y} p(y | \hat{u}) = E(y | \hat{u}).
\]

By \(p(z, u, y)\) we mean the joint density for all three random vectors, and by \(p(y | \hat{u})\) we mean the density for \(y\) given that \(u = \hat{u}\).

\textbf{Proof:} We have

\[
p(z, u, y) = p(z | u, y) p(u, y) = p(z | u, y) p(y | u) p(u) = p(z | u) p(u) p(y | u).
\]

It follows from the definition of \((\hat{u}, \hat{y})\), and the last equation above, that

\[
\hat{y} = \arg\max_{y} p(y | \hat{u}) = \arg\max_{y} p(z, \hat{u}, y),
\]

which proves the second assertion of this lemma. Define

\[
\bar{u} = \arg\max_{u} p(z | u) p(u), \quad \bar{y} = \arg\max_{y} p(y | \bar{u}).
\]

Using (4), Lemma 2.1, and (5) we have

\[
p(z, \bar{u}, \bar{y}) = p(z | \bar{u}) p(\bar{u}) p(\bar{y} | \bar{u}) = p(z | \bar{u}) p(\bar{u}) \underbrace{p(\bar{y} | \bar{u})}_{1} \leq p(z, \bar{u}, \bar{y})
\]

The reverse inequality follows from the definition of \((\hat{u}, \hat{y})\). Thus equality holds and it follows from the uniqueness assump-
tion that \( \hat{u} = \tilde{u} \). This proves the first assertion of the lemma and thereby completes the proof. \( \square \)

Remark 2.3. For some applications of Lemma 2.2, there is a deterministic matrix \( Y \) such that \( y = \hat{Y}u \). In this case, \( \hat{p}(y|u) \) is a delta function concentrated at \( u \) and \( \hat{y} = \hat{Y}u \). For other applications, there is a non-zero deterministic matrix \( U \) such that \( u = YU \). In this case, the matrix \( V(y) - V(y,u)V(u, y) \) in Lemma 2.1 is singular and
\[
\hat{y} = \arg\max_y p(y) \text{ subject to } \hat{u} = \hat{Y}y.
\]

Fix a sequence of time points \( \{t_1, \ldots, t_N\} \) and define
\[ x_i = x(t_i), \quad H_i = H(t_i), \quad f_i = f(t_i) = H_i x_i, \quad x_i^{i+1} = (x_i, t_i, x_i^{i+1}) \]
To simplify the notation, we set \( Y = 1 \) in (1). For any function \( v : \mathbb{R} \rightarrow \mathbb{R}^d \), we use the notation \( v(\cdot) \) for the function and \( v = [v(t_1)^T, v(t_2)^T, \ldots, v(t_N)^T]^T \) for the sample vector \( v \in \mathbb{R}^{dN} \). In addition, we use the notation \( \mathbb{H}(\cdot) \) for the measurement function and \( \mathbb{H} \in \mathbb{R}^{N 	imes N} \) for the matrix
\[
\mathbb{H} = \begin{bmatrix}
H_1 & 0 & \cdots & 0 \\
0 & H_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & H_N
\end{bmatrix}
\]
The following proposition states the relationship between Kalman smoothing and regularization in RKHS.

Proposition 2.4. Let \( f(\cdot) \) be a zero-mean Gaussian process with prior distribution given by (3). \( \hat{p} : \mathbb{R} \rightarrow \mathbb{R} \) be a loss function such that
\[
\hat{p}(z|f) \propto \prod_{i=1}^N \exp \left[ -\frac{1}{2} \rho(z_i - f_i) \right].
\]
Let \( \hat{c} \) be the maximum a posteriori (MAP) estimate of \( x \) conditional on \( z \), i.e.
\[
\hat{c} = \arg\min_z \sum_{i=1}^N \rho(z_i - H_i x_i) + \frac{1}{2} \Sigma^{-1} x,
\]
where \( \Sigma \) is the covariance of the state vector \( x \) and \( \Sigma^{-1} \) is block tridiagonal. Then, the solution of (1) is given for any \( \tau \in \mathbb{R} \) by
\[
\hat{f}(\tau) = \begin{bmatrix}
H_1 \hat{x}_1 \\
H(\tau) \mathbb{V}(x(\tau), x_i^{i+1}) \mathbb{V}(x_i^{i+1})^{-1} \hat{x}_i^{i+1}, \quad \tau < t_i \\
H(\tau) \mathbb{V}(x(\tau), x_i^{i+1}) \mathbb{V}(x_i^{i+1})^{-1} \hat{x}_i, \quad \tau < t_i \\
H(\tau) \mathbb{V}(x(\tau), x_N) \mathbb{V}(x_N)^{-1} \hat{x}_N, \quad t_N > \tau
\end{bmatrix},
\]
where \( i \) is understood to be in \( \{1, \ldots, N\} \) where it appears in the expression above.

Proof: The random vectors \( f \in \mathbb{R}^N \) and \( x \in \mathbb{R}^{N} \) are jointly Gaussian, because \( f = Hx \). In addition \( \hat{p}(z|f,x) = \hat{p}(z|f) \). Hence we can apply Lemma 2.2 with \( u = f \) and \( y = x \) (see Remark 2.3). This lemma defines \( \hat{f} \) and \( \hat{c} \) by
\[
[\hat{f}, \hat{c}] = \arg\max_{f,x} \hat{p}(z, f, x)
\]
Using \( f = Hx \), we can remove \( f \) from the optimization problem above and get
\[
\hat{f}(\tau) = \arg\max_x \hat{p}(x|\hat{f}, \tau) = \arg\max_x \hat{p}(x|\hat{f}) \cdot \hat{p}(f|\hat{f}) = \hat{f}(\tau).
\]
This shows that \( \hat{f} \) is the MAP estimator for \( x \). Hence we can complete the proof by showing that (8) is true. We begin by showing the \( \hat{f} = \hat{f} \). According to Lemma 2.2 we also have
\[
\hat{f}(\tau) = \arg\max_x \hat{p}(x|\hat{f}) \cdot \hat{p}(f|\hat{f})
\]
The function \( K(\cdot, \cdot) \) is the covariance for \( f(\cdot) \) and \( K \in \mathbb{R}^{N \times N} \) the autocovariance of the sample vector \( f \in \mathbb{R}^N \). Taking twice the negative log of the right hand side in the equation above gives
\[
\hat{f}(\tau) = \arg\min_{f} \sum_{i=1}^N \rho(z_i - f_i) + f^T K^{-1} f.
\]
Define \( c = K^{-1} f \). It follows that \( \hat{f} = Kc \) where \( c = \hat{c} \) and hence \( \hat{f} = \hat{f} \). Thus we have proved the following assertion:

Assertion: Equation (8) is true for \( \tau \in \{t_1, \ldots, t_N\} \).

Consider the case where \( \tau \notin \{t_1, \ldots, t_N\} \). The prior for \( f(x) \in \mathbb{R}^N \) is jointly Gaussian. In addition we have
\[
\hat{p}(z|x, f(\tau)) = \hat{p}(z|x) \quad \text{and} \quad [\hat{f}, \hat{c}(\tau)] = \arg\max_{f, \hat{c}} \hat{p}(z, f, x). \]

Thus we can apply Lemma 2.2 with \( u = x \) and \( y = f(\tau) \) to conclude that
\[
\hat{f}(\tau) = \arg\max_{f(\tau)} \hat{p}(f(\tau)|\hat{c}) = \hat{E}(f(\tau)|\hat{c}) = H(\tau) E[x(\tau)|\hat{c}].
\]
In addition, we can apply Lemma 2.2 with \( u = f \) and \( y = f(\tau) \) to conclude that
\[
\hat{f}(\tau) = \arg\max_{f(\tau)} \hat{p}(f(\tau)|\hat{f}) = \arg\max_{f(\tau)} \hat{p}(f(\tau) | \hat{f}) = \hat{f}(\tau).
\]
Note that the definition for \( \hat{f}(\tau) \) in this application of Lemma 2.2 has the same value as when \( u = x \) and \( y = f(\tau) \). Also note that \( \hat{f} = \hat{f} \) was shown by the assertion above. Finally note that this sequence of equalities shows the following assertion:

Assertion: \( \hat{f}(\tau) = \hat{f}(\tau) \) for all \( \tau \).

We now restrict our attention to the case where there is an \( i \in \{1, \ldots, N-1\} \) such that \( t_i < \tau < t_i+1 \). Using the Markov property, we have
\[
\hat{E}(x(\tau)|\hat{c}) = \hat{E}(x(\tau)|x_i^{i+1}) = V[x(\tau), x_i^{i+1}]V(x_i^{i+1})^{-1} x_i^{i+1},
\]
\[
\hat{f}(\tau) = \hat{f}(\tau) = H(\tau) V[x(\tau), x_i^{i+1}]V(x_i^{i+1})^{-1} x_i^{i+1}.
\]
This proves the assertion in equation (8) for the case where \( t_i \leq \tau \leq t_{i+1} \). The other two cases \( \tau < t_i \) and \( t_{i+1} < \tau \) can be obtained following the same reasoning as in the paragraph above. This completes the proof. \( \square \)
The above result shows that given  hat{l}, the solution to (1) can be computed using equation (8). In other words, the estimates of the state at sampling locations are sufficient to reconstruct the function hat{f} (·) over its entire domain. Moreover, the optimization problem (7) exploits the block triadiagonal structure of Σ^−1.

In the case where ρ(·) is quadratic, (De Nicolao and Ferrari Treocate, 2001, Theorem 1) provides a procedure for obtaining the weight vector  hat{l} at the same time as obtaining the Kalman smoother estimate  hat{x}. That procedure has O(Nn^3) computational complexity. In the sections below we present an interior point algorithm that obtains the same complexity when ρ(·) is the Vapnik loss.

3. KALMAN SMOOTHING WITH THE VAPNIK PENALTY

Here and below, we assume −log ρ(z|x) in (6) is the Vapnik loss function. The Vapnik loss function with parameter ε, also known as the ε-insensitive loss function, is displayed in Figure 1 and given by

\[ ρ(y) = \begin{cases} y - ε & \text{if } y ≥ ε \\ 0 & \text{if } y ∈ [-ε, +ε] \\ y + ε & \text{if } y ≤ -ε \end{cases} \]

Applying Proposition 2.4 to the Vapnik case, the MAP objective as a function of  x ∈ R^{Nn} is

\[ \frac{1}{2} \Sigma^{-1} x + \sum_{i=1}^{N} \left[ \sup_{u^+(i) ∈ [0,1]} u^+_i (z_i - H_i x_i - ε) + \sup_{u^-(i) ∈ [0,1]} u^-_i (z_i - H_i x_i - ε) \right] \]

where  u^+_i ∈ R^N,  u^-_i ∈ R^N, and  u^+(i) = u^+_i,  u^-(i) = u^-_i are used to avoid two levels of subscripting. Note that this objective is strictly convex, and so has a unique global minimum.

Lemma 3.1. Suppose that  x ∈ R^{Nn} and there are vectors  u^+,  u^- s^+, s^- p^+, p^- q^+, q^- all in  R^N, such that the following conditions hold for  i = 1, N:

\[ \Sigma^{-1} x - H^T (u^+ - u^-) = 0 \]

It follows that this choice of  x above minimizes (9) over  R^{Nn}.

Proof: For a convex set  A ⊂ R, and  a ∈ A, we use  N(A,a) to denote the normal cone to  A at the point  a; e.g.,

\[ N([0,1],a) = \begin{cases} (-∞,0] & \text{if } a = 0 \\ \{0\} & \text{if } a ∈ (0,1) \end{cases} \]

See (Rockafellar, 1970) for more details on normal cones in general. The optimality conditions for minimizing (9) are condition (10) and

\[ z_i - H_i x_i - ε ∈ N ([0,1], u^+_i) \quad (i = 1, \cdots N) \]

\[ -z_i + H_i x_i - ε ∈ N ([0,1], u^-_i) \quad (i = 1, \cdots N). \]

See (Rockafellar and Wets, 1998) for a discussion of these optimality conditions. We will show that condition (11) implies condition (13). The fact that condition (12) implies condition (14) can be demonstrated in a similar manner and hence this will complete the proof.

The optimality condition in (13) is equivalent to the following three conditions for  i = 1, N:

\[ z_i - H_i x_i - ε < 0 \Rightarrow u^+_i = 0 \]

\[ z_i - H_i x_i - ε = 0 \Rightarrow u^+_i ∈ [0,1] \]

\[ z_i - H_i x_i - ε > 0 \Rightarrow u^-_i = 1 \]

Hence it will suffice to show that the conditions (11) imply the three conditions above. We divide this demonstration into the three cases corresponding to the three conditions above:

1. Suppose that  z_i - H_i x_i - ε < 0: It follows from (11) that  q^+_i > 0 and  u^-_i = 0. Hence (15) holds.

2. Suppose that  z_i - H_i x_i - ε = 0: It follows from (11) that  u^-_i ∈ [0,1]. Hence (16) holds.

3. Suppose that  z_i - H_i x_i - ε > 0: It follows from (11) that  p^-_i > 0,  s^-_i = 0, and  u^-_i = 0. Hence (17) holds.

This completes the proof of this lemma.

We minimize (9) using an interior point method (see Kojima et al. (1991); Nemirovskii and Nesterov (1994)). The main idea is to relax the complementarity conditions in (11) and (12) to obtain (18) below, and solve this system using a damped Newton’s method. The relaxation parameter  μ is driven to 0, yielding the unique solution to (9). All of the implementation details are given below.

Define 1x ∈ R^N to be the vector with all of its components equal to one. Given a  v^+ ∈ R^N ( v^- ∈ R^N) define  V_+ ∈ R^{N×N} ( V_- ∈ R^{N×N}) to be the diagonal matrix with with diagonal  v^+ ∈ R^N ( v^- ∈ R^N). This is correspondence between lower case and upper case letters applies only to vectors with the superscript plus or minus. Fix a relaxation parameter  μ and define  F_μ : R^{N+nN} → R^{N+nN} by

\[ F_μ = \begin{bmatrix} p^+ & q^+ & u^+ & s^+ \\ q^- & u^- & s^- & x \\ Q_u - 1_N & -μ 1_N & μ 1_N & 0 \\ P_s - 1_N & 0 & μ 1_N & -μ 1_N \end{bmatrix} \]

The last line in the definition of  F_μ is the residual equation (10). Expressions  r_2, r_3, r_6 and  r_7 in the definition of  F_μ are μ-relaxed versions of the residuals in the first two equations of (11) and (12), while  r_1, r_3, r_5, and  r_7 are the residuals in the last two
equations of (11) and (12). Thus if $u^+, u^- s^+, s^- p^+, p^- q^+, q^-$ are all in $\mathbb{R}^N_+$, and
\[
F_\mu(p^+, q^+, u^+, s^+, p^-, q^-, u^-, s^-) = 0,
\]
then $x$ is the unique minimizer of (9). Solutions to the equation $F_\mu = 0$ in (18) can be obtained via a damped Newton’s method by solving
\[
\begin{bmatrix}
I & -I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & H \\
0 & U_+ & Q_+ & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
S_+ & 0 & 0 & P_+ & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & -H \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & P_+ & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & H^T & 0 \\
\end{bmatrix}
\begin{bmatrix}
\Delta p^+ \\
\Delta q^+ \\
\Delta u^+ \\
\Delta s^+ \\
\Delta p^- \\
\Delta q^- \\
\Delta u^- \\
\Delta s^- \\
\end{bmatrix}
= \begin{bmatrix}
r_1 \\
r_2 \\
r_3 \\
r_4 \\
r_5 \\
r_6 \\
r_7 \\
r_8 \\
\end{bmatrix}
\]  
(19)
where the matrix on the left hand side of (19) is $F_\mu^{(1)}$.

**Lemma 3.2.** We are given values for the variables $p^+, q^+, u^+, s^+, p^-, q^-, u^-, s^-$, all in $\mathbb{R}^N$, $x$ in $\mathbb{R}^N$, and the parameters $\mu, e \in \mathbb{R}^+, \epsilon \in \mathbb{R}^N, H \in \mathbb{R}^{N \times N}$, and $\Sigma \in \mathbb{R}^{N \times N}$.

The algorithm (20) below computes the values $\Delta p^+, \Delta q^+$, $\Delta p^-, \Delta q^-$, $\Delta u^+, \Delta s^+$, $\Delta u^-, \Delta s^-$, all in $\mathbb{R}^N$ and $\Delta x \in \mathbb{R}^N$ that solve equation (19). Note $\{r_i\}, i = 1, \ldots, 9$, are defined in (18).

\[
\begin{align*}
T_+ & = S_+^3 P_+ + U_+^{-1} Q_+ \\
T_- & = S_-^3 P_+ + U_+^{-1} Q_+ \\
f_4 & = S_+^3 s_+ - r_1 - U_+^{-1} r_2 + U_+^{-1} Q_+ + r_3 \\
f_8 & = S_-^3 s_+ - r_5 - U_+^{-1} (r_6 - q_7 - r_7 + T_+^{-1} f_8) \\
\Delta x & = \Sigma^{-1} + H^T (T_+^{-1} + T_-^{-1}) H^{-1} f_9 \\
\Delta u^- & = T_-^{-1} (f_8 - H \Delta x) \\
\Delta u^+ & = r_7 - \Delta s^- \\
\Delta q^+ & = U_-^{-1} (r_6 - Q \Delta u^-) \\
\Delta p^+ & = r_5 + \Delta q^+ + H \Delta x \\
\Delta u^- & = r_7 - \Delta s^- \\
\Delta q^- & = U_-^{-1} (r_6 - Q \Delta u^-) \\
\Delta p^- & = r_5 + \Delta q^- + H \Delta x \\
\Delta s^+ & = T_+^{-1} (f_4 + H \Delta x) \\
\Delta s^- & = T_-^{-1} (f_4 + H \Delta x) \\
\Delta p^+ & = r_1 + \Delta q^+ + H \Delta x .
\end{align*}
\]  
(20)

**Proof:** We now provide row operations necessary to reduce the matrix $F_\mu^{(1)}$ to block upper triangular form. In addition, we track the changes to the right hand side in (19). Note that only rows 4, 8, and 9 of $F_\mu^{(1)}$ need to be modified to implement the reduction. To save space, after a series of row operations, we present the final row and corresponding right hand side. We use the symbol $\leftarrow$ to indicate assignment:

\[
\begin{align*}
\text{row}_4 & \leftarrow \text{row}_4 - S_+ \text{row}_1 \\
\text{row}_8 & \leftarrow \text{row}_8 - S_+ U_+^{-1} \text{row}_2 \\
\text{row}_4 & \leftarrow \text{row}_4 + S_+ U_+^{-1} Q_+ \text{row}_3 \\
\text{row}_4 & \leftarrow S_-^{-1} \text{row}_4
\end{align*}
\]

The final result for row 4 and the corresponding right hand side, which we label $f_4$, are given by

\[
\begin{align*}
\text{row}_4 & = \begin{bmatrix} 0 & 0 & 0 & T_+ & 0 & 0 & 0 & 0 & -H \end{bmatrix} \\
f_4 & = S_+^3 r_4 - r_1 - U_+^{-1} r_2 + U_+^{-1} Q_+ + r_3,
\end{align*}
\]
where $T_+ \in \mathbb{R}^{N \times N}$ is defined by $T_+ = S_+^{-1} P_+ + U_+^{-1} Q_+$.

The reduction of row 8 is analogous:

\[
\begin{align*}
\text{row}_8 & \leftarrow \text{row}_8 - S_- \text{row}_5 \\
\text{row}_8 & \leftarrow \text{row}_8 - S_- U_-^{-1} \text{row}_6 \\
\text{row}_8 & \leftarrow \text{row}_8 + S_- U_-^{-1} Q_+ \text{row}_7 \\
\text{row}_8 & \leftarrow S_-^{-1} \text{row}_8
\end{align*}
\]

The final form for row 8 and the corresponding right hand side, which we label $f_8$, are given by

\[
\begin{align*}
\text{row}_8 & = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & T_+ \end{bmatrix} \\
f_8 & = S_-^{-1} r_8 - r_5 - U_-^{-1} r_6 + U_-^{-1} Q_+ \text{row}_7 \\
\end{align*}
\]
where $T_- \in \mathbb{R}^{N \times N}$ is defined by $T_- = S_-^{-1} P_+ + U_-^{-1} Q_+$.

The final modifications are to row 9:

\[
\begin{align*}
\text{row}_9 & \leftarrow \text{row}_9 + H^T \text{row}_3 \\
\text{row}_9 & \leftarrow \text{row}_9 - H^T T_+^{-1} \text{row}_4 \\
\text{row}_9 & \leftarrow \text{row}_9 - H^T T_-^{-1} \text{row}_8 \\
\end{align*}
\]

The final forms of rows 9 and the corresponding right hand side, which we label $f_9$, are given by

\[
\begin{align*}
\text{row}_9 & = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Sigma^{-1} + H^T (T_+^{-1} + T_-^{-1}) H \end{bmatrix} \\
f_9 & = r_9 + H^T r_3 - H^T T_+^{-1} f_4 - H^T r_7 + H^T T_-^{-1} f_8
\end{align*}
\]

We now have a block upper triangular system which is equivalent to (19). In the remaining steps, we solve this system for $\Delta x, \Delta s^-, \Delta u^-, \Delta q^-, \Delta p^-, \Delta s^+, \Delta u^+, \Delta q^+, \Delta p^+$, and the equations $T_+^{-1} f_4, f_8, f_9$ provided above comprise the algorithm (20).

This completes the proof of the lemma. □

The corresponding system of equations is block upper triangular, and its solution depends on inversion of the matrices $U_+, U_-, S_+, S_-, T_+, T_-$, and $\Sigma^{-1} + H^T (T_+^{-1} + T_-^{-1}) H$.

The relaxed complementarity conditions in (18) are

\[
\begin{align*}
Q_+ U_+ t_1 & = \mu_1 \mathbb{1}_N \\
P_+ S_+ t_2 & = \mu_1 \mathbb{1}_N \\
Q_- U_- t_1 & = \mu_1 \mathbb{1}_N \\
P_- S_- t_2 & = \mu_1 \mathbb{1}_N 
\end{align*}
\]

For $\mu > 0$, solutions of the equation above are strictly positive; i.e., the solutions satisfy

\[
(p^+, q^+, u^+, s^+, p^-, q^-, u^-, s^-) > 0.
\]

Primal-dual interior point methods use predictor and corrector steps to follow the solution of $F_\mu = 0$ as $\mu > 0$ descends to zero. These methods keep the vectors $u^+, u^-, s^+, s^-$, and $s^+$ strictly positive. Hence the diagonal matrices $U_+, U_-, S_+ and S_-$ can be easily inverted. The vectors $p^+$ and $q^+$ are also strictly positive, so the diagonal matrices

\[
T_+ = S_+^{-1} P_+ + U_+^{-1} Q_+
\]
can be easily inverted. In addition, the vectors $p^-$ and $q^-$ are strictly positive, so the diagonal matrix

\[
T_- = S_-^{-1} P_+ + U_-^{-1} Q_-
\]
can be easily inverted. Finally, consider inverting the matrix

\[
T = \Sigma^{-1} + H^T (T_+^{-1} + T_-^{-1}) H.
\]
The \( nN \times nN \) matrix \( \Sigma^{-1} \) is block tridiagonal with blocks of size \( n \times n \). It can be inverted in \( O(n^3N) \) operations using (Bell, 2000, Lemma 6). Furthermore, the \( nN \times nN \) matrix \( H^T (T_n^{-1} + T_{-n}^{-1}) H \) is block diagonal with positive semi-definite blocks of size \( n \times n \). It follows that \( T \) can also be inverted in \( O(n^3N) \) operations using (Bell, 2000, Lemma 6). Thus, notably, the computational complexity of the Vapnik based smoother is linear in the number of time steps, as that of the classical Kalman smoother that relies upon quadratic losses.

4. NUMERICAL EXAMPLE

In this section we test the new Kalman smoother that incorporates the Vapnik’s loss via a simulated example. The unknown function, taken from (Dinuzzo et al., 2007), is given by

\[
 f(t) = \epsilon \sin(4t) \]

and has to be reconstructed from 2000 noisy samples collected uniformly over the unit interval. The measurement noise \( v_k \) was generated using a mixture of two normals with \( p = 0.1 \) denoting the fraction from each normal; i.e.,

\[
 v_k \sim (1 - p)N(0, 0.25) + pN(0, 25).
\]

Data are displayed as dots in Fig. 2. Note that the purpose of the second component of the normal mixture is to simulate outliers in output data. Note also that any points exceeding vertical axis limits are plotted at Fig. 2 to improve readability.

The initial condition \( f(0) = 1 \) is assumed to be known, while the difference of the unknown function from the initial condition (i.e. \( f'(0) - 1 \)) is the second component of the two-dimensional state vector \( x(t) \) and corresponds to the integral of the first state component, which is modeled as Brownian motion. To be more specific, letting \( \Delta t = 1/2000 \), the process model for the mean of \( x_k \) given \( x_{k-1} \) is

\[
 F_k(x_{k-1}) = \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} x_{k-1},
\]

while the autocovariance of \( x_k \) given \( x_{k-1} \) is

\[
 Q_k = \lambda^2 \begin{bmatrix} \Delta t & \Delta t^2/2 \\ \Delta t^2/2 & \Delta t^3/3 \end{bmatrix},
\]

(Jazwinski, 1970; Oksendal, 2005), where \( \lambda^2 \) is an unknown scale factor to be estimated from the data. It corresponds to the inverse of the regularization parameter \( \gamma \) that appears in (1). The measurement model for the mean of \( z_k \) given \( x_k \) is

\[
 h_k(x_k) = (0, 1) x_k = x_{2,k},
\]

where \( x_{2,k} \) denotes the second component of \( x_k \).

In order to estimate the two unknown parameters \( \lambda \) and \( \epsilon \) characterizing the Vapnik loss, the 2000 measurements were randomly split into training and validation sets of 1300 and 700 data points, respectively. For each value of \( \lambda^2 \) and \( \epsilon \) contained in a \( 10 \times 20 \) grid on \([0.01, 10000] \times [0, 1] \), with \( \lambda^2 \) logarithmically spaced, the function estimate was rapidly obtained by the new smoother applied to the training set. Then, the relative average prediction error on the validation set was computed, see Fig. 3. The parameters leading to the best prediction were \( \lambda^2 = 2.15e3 \) and \( \epsilon = 0.45 \), which give a sparse solution defined by less of 400 support vectors. For the sake of comparison, we also derived the solution of (1) using a quadratic loss function (by implementing a classical Kalman smoother) with \( \gamma \) estimated following the same validation strategy described above. Differently from the Vapnik penalty, the quadratic loss does not induce any sparsity, so that, in this case, the number of support vectors equals the size of the training set.

The left and right panels of Fig. 2 report the function estimate adopting the quadratic and the Vapnik loss, respectively. It is clear that the Gaussian estimate is heavily affected by the outliers. In contrast, the estimate coming from the Vapnik based smoother performs well over the entire time period, being virtually unaffected by the presence of large outliers.

5. CONCLUSIONS

We have described an approach to solve a large class of machine learning, named state space kernel machines, and have focused on the non-smooth Vapnik penalty as the loss function of interest. In practise, the Vapnik loss function has two desirable features illustrated in the numerical experiment: robustness to outliers in the measurement data, and sparsity in the final representation of the reconstructed function. The interior point approach we use, together with Proposition 2.4, allows the development of learning machine algorithms for such losses with a time complexity that scales linearly with the number of samples. In the near future, we plan to extend the obtained results to more general losses, such as the soft \( \epsilon \)-insensitive and the Huber’s one.

ACKNOWLEDGEMENTS

This research has been partially supported by the PRIN Project “Sviluppo di nuovi metodi e algoritmi per l’identificazione, la stima Bayesiana e il controllo adattativo e distribuito”, by the Progetto di Ateneo CPDA090135/09 funded by the University of Padova and by the European Community’s Seventh Framework Programme under agreement n. FP7-ICT-223866-FeedNetBack.

REFERENCES


Fig. 2. Simulation: measurements (·) with outliers set to 4 or −2, true function (continuous line), smoothed estimate using either the quadratic loss (dashed line, left panel) or the Vapnik’s ε-insensitive loss (dashed line, right panel)


