A PARTICLE FILTERING BASED KERNEL HMM PREDICTOR

Felipe A. Tobar and Danilo P. Mandic

Department of Electrical and Electronic Engineering
Imperial College London, UK

ABSTRACT

A novel kernel algorithm is proposed for nonlinear prediction whereby the signal is modelled as a state of a hidden Markov model (HMM). The transition function of the HMM is approximated using kernels, whose weights are also part of the state of the system and are learnt in an unsupervised fashion by a sample importance resampling (SIR) particle filter. The SIR proposal density is designed so as to maintain a diverse population of particles, thus avoiding particle degeneracy arising from inaccuracies of early model estimates. The kernel HMM algorithm is further equipped with a sparsification criterion based on approximate linear dependence and its performance is evaluated against the KNLMS and KRLS algorithms for the prediction of synthetic signals and real world point-of-gaze data.

Index Terms— Kernel LMS, kernel RLS, hidden Markov models, particle filters, gaze tracking.

1. INTRODUCTION

The prediction problem refers to the estimation of the process X_{t+1} given the observation[1] path $Y_{0:t} = \mathbf{y}_{0:t}$ [1]. When noisy observations of the process $X_{0:t} = \mathbf{x}_{0:t}$ are available, X_{t+1} can be predicted using a supervised regression model. These include least squares or ridge regression for the batch processing case, and least mean square (LMS) or recursive least squares (RLS) for the recursive mode [2]. Alternatively, when $X_{0:t}$ is a hidden process with known evolution and observation dynamics, the Kalman filter (linear case) and its nonlinear extensions, as well as particle methods, are a *de facto* standard for signal filtering and prediction [3–5].

Standard prediction algorithms do not cater for model uncertainty, as they either assume linear models or require complete knowledge of the underlying latent dynamics using e.g. a hidden Markov model (HMM). To enable prediction even in cases when the system model is known only partially, we propose an additional degree of freedom which enables the filter to *learn* the HMM functions using reproducing kernels [6], a class of universal function approximators which can be trained to replicate the relationship between input-output pairs. Kernel methods have been successfully applied in nonlinear filtering based on the so-called *kernel trick* [7] and provide robust nonlinear estimation with low computational complexity. Established algorithms include the kernel ridge regression [8], kernel recursive least squares (KRLS) [9], kernel least mean square (KLMS) [10, 11], kernel affine projection and kernel normalised LMS (KNLMS) [12], and multikernel LMS [13]. These have found applications in channel equalisation, wind prediction, trajectory tracking, and fault diagnosis [14–19].

A major limitation of the existing kernel prediction approaches is that they provide point estimates rather than the complete prediction density. We rectify this issue by modelling the signal as the state of an HMM and using kernels to learn the state-transition function of the HMM, thus allowing for a full statistical description of the process. Since the state of the HMM is—by definition—unobservable, this requires unsupervised learning. We base our approach on self-organising state space models [20], whereby the unknown parameters are considered part of the state of the HMM and are then *filtered* jointly with the original state from the observations using particle filters.

Previous research combining HMM and kernel regression includes the approaches in [21], which identifies a system by kernel estimation, and in [22], which represents the distribution of the model using points in a Hilbert space. Both of these algorithms are supervised and trained with noisy observations, making them unsuitable for non-stationary environments. Unsupervised approaches include kernel density estimation of the model likelihoods using Markov chain Monte Carlo (MCMC) in the context of multivariate variance estimation [23], and reversible-jump MCMC methods [24] that are flexible and robust to the choice of priors owing to their hierarchical structure. These, however, require a fair amount of information to converge, making them inadequate for online applications.

This all highlights the void in the open literature concerning a class of algorithms that brings under one umbrella: (i) the ability of kernel adaptive filters to jointly train and predict signals online; (ii) the stochastic modelling strength of HMMs; and (iii) the suitability of particle filters for nonlinear filtering. To this end, we introduce a particle filtering based unsupervised kernel HMM algorithm, and evaluate its performance against both the KNLMS and KRLS for the prediction of synthetic nonlinear signals and gaze tracking. The analysis shows that kernel HMM predictions are not only as accurate as those produced by the existing KNLMS and KRLS, but are also robust to excessive levels of observation noise. In addition, we show that the second order statistics and distribution of the kernel HMM predictions are consistent with the observed signals, and can therefore be used to compute reliable confidence intervals.

2. KERNEL-BASED PREDICTION

Consider the discrete-time process $\{X_t\}_{t\in\mathbb{N}}$ defined by

$$X_{t+1} = f_t(X_t) + V_t, \tag{1}$$

where $\{V_t\}_{t\in\mathbb{N}}$ is a noise process, $f_t(\cdot)$ a nonlinear function, and observations $X_t = \mathbf{x}_t$ become available sequentially.

At time t, the value of X_{t+1} can be predicted via kernel regression [25] in the form

$$\widehat{X}_{t+1} = \sum_{j=1}^{N} \omega_{t,j} K(\mathbf{x}_t, \mathbf{s}_j) \tag{2}$$

where K is a reproducing kernel [6], $\{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_N\}$ are the support vectors, and $\boldsymbol{\omega}_t = [\omega_{t,1}, \ldots, \omega_{t,N}]$ are the kernel weights at time

[1]Where Y_t denotes the random variable and $\mathbf{y}_t$ its realization at time t.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

instant t. This formulation provides nonlinear regression capability and requires finding a suitable set of support vectors and coefficients $\boldsymbol{\omega}_t$.

As the number of support vectors rapidly becomes overwhelming for online applications, sparsification procedures are needed to avoid excessive computational cost. We next review the sparsification criterion based on approximate linear dependence for choosing the support vectors.

2.1. Approximate linear dependence (ALD)

In the context of kernel regression, the ALD sparsification criterion [26] includes the observation $\mathbf{x}_t$ to the dictionary $D = \{\mathbf{s}_i\}_{i=1:N}$ when its feature sample $\phi(\mathbf{x}_t)$ does not fulfil the condition

$$\delta = \min_{\mathbf{a}\in\mathbb{R}^N} \|[\phi(\mathbf{s}_1), \dots, \phi(\mathbf{s}_N)]\,\mathbf{a} - \phi(\mathbf{x}_t)\|^2 \leq \eta \qquad (3)$$

for $\eta > 0$.

The optimal coefficients are calculated as $\mathbf{a}_{\text{opt}} = \mathbf{K}^{-1}\mathbf{h}(\mathbf{x}_t)$, where the entries of both the Gram matrix $\mathbf{K}$ and the kernel-evaluation vector $\mathbf{h}(\mathbf{x}_t)$ are given respectively by $\mathbf{K}_{p,q} = K(\mathbf{s}_p, \mathbf{s}_q)$ and $\mathbf{h}_p(\mathbf{x}_t) = K(\mathbf{x}_t, \mathbf{s}_p)$. Upon replacing $\mathbf{a} = \mathbf{a}_{\text{opt}}$ into (3), the ALD condition becomes

$$\delta = K(\mathbf{x}_t, \mathbf{x}_t) - \mathbf{h}^T(\mathbf{x}_t)\mathbf{a}_{\text{opt}} \leq \eta. \qquad (4)$$

The idea underpinning ALD can be summarised as follows: if a feature sample is *approximately linearly dependent* with respect to the current dictionary, its inclusion would be redundant and the associated computational cost would not justify the (marginal) increase in performance.

2.2. Kernel least mean square (KLMS)

The KLMS algorithm updates the coefficients of the prediction in Eq. (2) in an LMS fashion whenever a new observation $\mathbf{x}_t$ becomes available [10]. When the dictionary is not modified, the parameters $\boldsymbol{\omega}_t$ are updated according to

$$\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t + \mu \mathbf{e}_{t+1}\mathbf{h}^T(\mathbf{x}_t)$$

where $\mu > 0$ is the learning rate and $\mathbf{e}_{t+1} = \mathbf{x}_{t+1} - \boldsymbol{\omega}\mathbf{h}(\mathbf{x}_t)$ is the prediction error. Note that when a new vector $\mathbf{x}_t$ is added to the dictionary, the initial value of the corresponding weight is set to

$$\omega_{t+1,j} = 0 + \mu \mathbf{e}_{t+1} K(\mathbf{x}_t, \mathbf{x}_t).$$

We will use a normalised version of this algorithm, the kernel normalised LMS (KNLMS) [12], in our simulations in Section 4.

2.3. Kernel recursive least squares (KRLS)

In a similar fashion, the KRLS [9] uses the RLS strategy to update the coefficients $\boldsymbol{\omega}_t$. When a new sample $\mathbf{x}_t$ is added to the dictionary, the update rule becomes

$$\boldsymbol{\omega}_{t+1} = \left[\boldsymbol{\omega}_t + \delta^{-1}\mathbf{e}_{t+1}\mathbf{a}^T,\ \delta^{-1}\mathbf{e}_{t+1}\right]$$

whereas when the dictionary is not updated we have

$$\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t + \frac{\mathbf{e}_{t+1}\boldsymbol{\omega}_t\mathbf{P}_t\mathbf{K}^{-1}}{1 + \boldsymbol{\omega}_t\mathbf{P}_t\boldsymbol{\omega}_t^T}.$$

The parameters $\mathbf{a}$ and δ are computed as in the ALD step from Eqs. (3) and (4), while the matrix $\mathbf{P}_t$ is the covariance matrix of the RLS estimate, see [9] for more details.

3. A KERNEL HIDDEN MARKOV MODEL

Unlike the KLMS and KRLS algorithms, which provide point predictions by updating the parameters $\boldsymbol{\omega}_t$ based on the prediction error, our aim is to provide statistical description by estimating the full prediction density, thus allowing for the estimation of higher order statistics and confidence intervals. To this end, we model the signal and its observation as the state and output of a HMM respectively, and then estimate the prediction density using nonlinear filters.

Consider the state space model of the form

$$\begin{aligned} X_{t+1} &= \sum_{j=1}^{N} \omega_{t,j} K(X_t, \mathbf{s}_j) + V_t & (5)\\ Y_t &= X_t + W_t & (6)\end{aligned}$$

where $\{X_t\}_{t\in\mathbb{N}}$, $\{Y_t\}_{t\in\mathbb{N}}$ are respectively the latent and observed processes, and $\{V_t\}_{t\in\mathbb{N}}$,$\{W_t\}_{t\in\mathbb{N}}$ are independent noise processes.

Observe that this is a simplified version of an HMM, whereby the state transition function in (5) is a weighted kernel combination, and the observation function in (6) is an identity. With this form of the state transition, the task of model identification is reduced to finding the coefficient $\boldsymbol{\omega} \in \mathbb{R}^{1\times N}$ and the support vectors $\{\mathbf{s}_j\}_{j=1:N}$. In addition, we equip the proposed framework with the ALD sparsification criterion [26] for choosing the support vectors from the stream of observations $\mathbf{y}_{0:t}$. The update of $\boldsymbol{\omega}_t$ is elaborated in the next section.

Remark 1. *The estimation setting in (5)-(6) offers enhanced modelling capability, since the use of universal kernels guarantees that the expression $\sum_{j=1}^{N} \omega_{t,j} K(X_t, \mathbf{s}_j)$ approximates any continuous function $f_t(X_t)$ with arbitrary precision [27].*

Remark 2. *The aim of the proposed model is to estimate the statistics of the process X_{t+1} conditional to the observations $Y_{0:t} = \mathbf{y}_{0:t}$ in a non-parametric manner, and not necessarily to identify the true underlying system.*

3.1. State space model

Within the above kernel setting, we consider the unknown kernel weights $\boldsymbol{\omega}_t$ to be part of the state of the HMM, so that its posterior distribution $p(\boldsymbol{\omega}_t|\mathbf{y}_{1:t})$ can be estimated using particle filters. This yields the state space model

$$\begin{aligned} X_{t+1} &= \sum_{j=1}^{N} \omega_{t,j} K(X_t, \mathbf{s}_j) + V_t,\ \ V_t \sim \mathcal{N}\left(0, \Sigma_X^2\right)\\ \omega_{t+1,j} &= \omega_{t,j} + \epsilon_t^{\omega},\ \ \epsilon_t^{\omega} \sim \mathcal{N}\left(0, \sigma_\omega^2\right) & (7)\\ Y_t &= X_t + W_t,\ \ W_t \sim \mathcal{N}\left(0, \Sigma_Y^2\right)\end{aligned}$$

where $[X_t; \boldsymbol{\omega}_t]$ is the state of the HMM, and $\Sigma_Y^2, \Sigma_X^2, \sigma_\omega^2$ are the covariances of the corresponding processes.

This approach for system identification, referred to as *self-organising state space models* [20], assumes artificial evolution dynamics and is well known to provide reasonable estimates of model parameters when using particle filters in real-world applications [28, 29].

Remark 3. *The model (7) offers two distinguishing advantages: (i) by allowing the parameters $\boldsymbol{\omega}_t$ to gradually change, the resulting time-varying model is suitable for nonstationary environments; (ii) by virtue of the parameters being part of the (extended) system state, their posterior density can be found using particle filters.*

3.2. Prediction using particle filters

To estimate the joint posterior of X_t and $\boldsymbol{\omega}_t$ in (7) conditional to the observed path $Y_{0:t} = \mathbf{y}_{0:t}$, we propose to use sequential importance resampling (SIR) particle filters [30]. In this way, $p(X_t, \boldsymbol{\omega}_t | \mathbf{y}_{0:t})$ is approximated by a weighted average of particles $(\mathbf{x}_t^{(j)}, \boldsymbol{\omega}_t^{(j)})$, for which the weights $\alpha_t^{(j)}$ are recursively calculated based on the likelihood of each particle [5]. Additionally, a Gaussian proposal density π that is sufficiently wide is considered so as to fully explore the state space and to avoid particle degeneracy,

Remark 4. *Notice that in our case the usual practice of choosing the proposal density π equal to the model prior is highly inappropriate, since the HMM in (7) might not be able to resemble the desired model before a sufficient number of samples have been processed.*

The marginal filtering density of the SIR is then given by

$$\hat{P}_N\left(\mathbf{x}_t, \boldsymbol{\omega}_t | \mathbf{y}_{1:t}\right) = \sum_{j=1}^{N_p} \alpha_t^{(j)} \delta_{\mathbf{x}_t^{(j)}, \boldsymbol{\omega}_t^{(j)}}\left(\mathbf{x}_t, \boldsymbol{\omega}_t\right) \quad (8)$$

and the SIR weight update by

$$\alpha_t^{(j)} \propto \alpha_{t-1}^{(j)} \frac{p\left(\mathbf{y}_t \middle| \mathbf{x}_t^{(j)}, \boldsymbol{\omega}_t^{(j)}\right) p\left(\mathbf{x}_t^{(j)}, \boldsymbol{\omega}_t^{(j)} \middle| \mathbf{x}_{t-1}^{(j)}, \boldsymbol{\omega}_{t-1}^{(j)}\right)}{\pi\left(\mathbf{x}_t^{(j)}, \boldsymbol{\omega}_t^{(j)} \middle| \mathbf{x}_{t-1}^{(j)}, \boldsymbol{\omega}_{t-1}^{(j)}, \mathbf{y}_{1:t}\right)} \quad (9)$$

For simplicity, we consider a proposal density π whereby the particle components $\mathbf{x}_t^{(j)}, \boldsymbol{\omega}_t^{(j)}$ are jointly Gaussian and independent given $\mathbf{x}_{t-1}^{(j)}, \boldsymbol{\omega}_{t-1}^{(j)}$, that is

$$\begin{aligned} \mathbf{x}_t^{(j)}, \boldsymbol{\omega}_t^{(j)} &\sim \pi\left(\mathbf{x}_t^{(j)}, \boldsymbol{\omega}_t^{(j)} \middle| \mathbf{x}_{t-1}^{(j)}, \boldsymbol{\omega}_{t-1}^{(j)}, \mathbf{y}_{1:t}\right) \quad (10) \\ &= \mathcal{N}\left(\mathbf{x}_{t-1}^{(j)}, \Sigma_{\mathbf{x}}\right) \mathcal{N}\left(\boldsymbol{\omega}_{t-1}^{(j)}, \Sigma_{\boldsymbol{\omega}}\right). \end{aligned}$$

Finally, the prediction distribution of the hidden process is calculated using the estimate of the posterior density in Eq. (8) to yield

$$\begin{aligned} \hat{p}(\mathbf{x}_{t+1} | \mathbf{y}_{1:t}) &= \int p(\mathbf{x}_{t+1} | \mathbf{x}_t, \boldsymbol{\omega}_t) \hat{P}_N(\mathbf{x}_t, \boldsymbol{\omega}_t | \mathbf{y}_{1:t}) \mathrm{d}\mathbf{x}_t \mathrm{d}\boldsymbol{\omega}_t \\ &= \sum_{j=1}^{N_p} \alpha_t^{(j)} p(\mathbf{x}_{t+1} | \mathbf{x}_t^{(j)}, \boldsymbol{\omega}_t^{(j)}) \quad (11) \end{aligned}$$

where $\mathrm{d}\mathbf{x}_t \mathrm{d}\boldsymbol{\omega}_t$ is the Lebesgue measure in $X \times \mathbb{R}^N$. Based on the priors in the model (7), the estimated prediction of the process X_t becomes

$$\mathbb{E}\{X_{t+1} | \mathbf{y}_{0:t}\} = \sum_{j=1}^{N_p} \alpha_t^{(j)} \sum_{i=1}^{N} \omega_{t,i}^{(j)} K(\mathbf{x}_t^{(j)}, \mathbf{s}_i). \quad (12)$$

Fig. 1 summarises the information flow within the proposed kernel HMM algorithm, including the components of the SIR particle filter, ALD sparsification, and prediction stage. The input (observation $\mathbf{y}_t$) and output (predictions) of the algorithm are respectively given in red and brown, while the particles are colour coded in magenta, the SIR weights in green, and the support vectors in blue.

4. SIMULATION RESULTS

The performance of the proposed kernel HMM was evaluated against the KNLMS [12] and KRLS [9] for the prediction of synthetic data and real-world two-dimensional point-of-gaze signals.

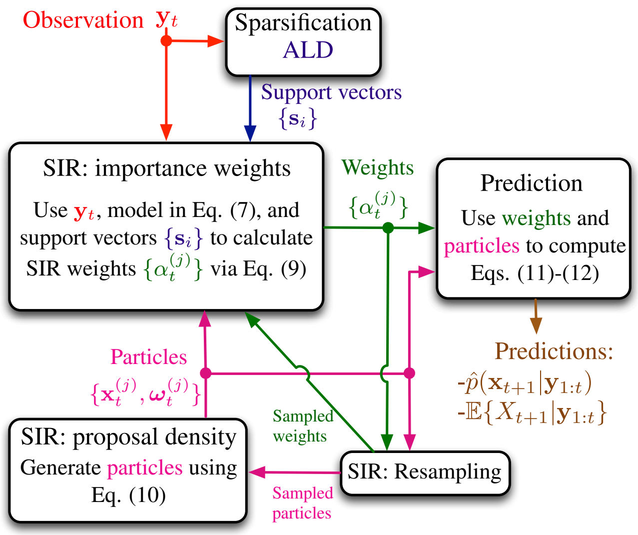

Fig. 1: Block diagram of proposed kernel HMM algorithm.

4.1. Nonlinear prediction

In the first set of simulations, the following benchmark nonlinear system [5] was considered

$$\begin{aligned} X_{t+1} &= \frac{X_t}{2} + \frac{25 X_t}{1 + X_t^2} + 8\cos(0.8t) + V_t \\ Y_t &= X_t + W_t \end{aligned}$$

where $V_t \sim \mathcal{N}(0,1)$ and $W_t \sim \mathcal{N}(0,9)$.

The goal was to recursively predict X_{t+1} from the noisy observations $Y_{0:t} = y_{0:t}$. The kernel width and ALD threshold were $A = 0.8$ and $\delta = 0.1$, respectively. The algorithm parameters were $\mu = 0.6$ for the KNLMS and the variances $\Sigma_X^2 = 20, \sigma_\omega^2 = .01, \Sigma_Y^2 = 9$ for the kernel HMM with $N_p = 400$ particles.

Fig. 2 shows the hidden process X_t, its kernel HMM prediction, and the confidence interval corresponding to two standard deviations centred about the prediction. Observe that the estimated two-standard-deviation confidence interval progressively included the original signal as more information became available, highlighting the statistical prediction ability of the proposed kernel HMM as opposed to standard point predictions. The average prediction mean squared error (MSE) of the kernel algorithms considered was evaluated over 50 independent realisations and is given in Table 1; the kernel HMM outperformed the KNLMS and KRLS. Observe that the better performance of KNLMS over KRLS indicates the nonstationarity of the system.

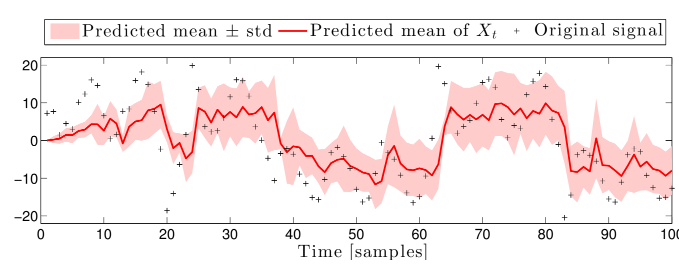

Fig. 2: Prediction using the proposed kernel HMM.

Table 1: Prediction performance of kernel algorithms

	Kernel HMM	KNLMS	KRLS
MSE	**8.54**	9.08	10.09

4.2. Tracking coordinates of eye movement

We next validated the kernel HMM for the task of gaze prediction. A Tobii T60 Eye Tracker was used to acquire the horizontal and vertical point-of-gaze signals when reading a 20-word text arranged in two lines. This is a challenging task, as a *linear* reading of the task is followed by a *jump* to the beginning of the next line.

The data were centered, missing values were replaced by their previous value and a preliminary recording was used to set the empirical parameters: a) kernel width $A = 5 \cdot 10^{-4}$, b) KNLMS gain $\mu = 0.6$, c) ALD threshold $\delta = 0.1$, d) number of particles $N_p = 800$ and e) kernel HMM variances $\sigma_\omega^2 = 1$,

$$\Sigma_X^2 = 300 \begin{bmatrix} 10 & 1 \\ 1 & 1 \end{bmatrix}, \; \Sigma_Y^2 = 300 \begin{bmatrix} 5 & 2.45 \\ 2.45 & 2.8 \end{bmatrix}.$$

The performances of all three kernel algorithms considered were assessed in the presence of additive Gaussian noise of different power added to the point-of-gaze data; the averaged performances as a function of the signal-to-noise ratio (SNR) are shown in Fig. 3. The kernel HMM (in red) proved to be more robust to uncertainties in the observations than the KNLMS and KRLS.

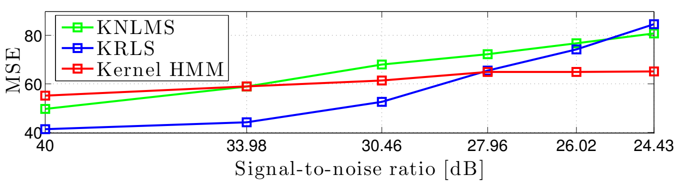

Fig. 3: Performances of kernel algorithms for gaze prediction.

Fig. 4 presents the measurements of the gaze signal and the kernel HMM predictions for the signal mean and standard deviation for the complete duration of the recording for the SNR=26.02 [dB], highlighting the suitability of the algorithm to track point-of-gaze changes as well as its accuracy. Fig. 5 illustrates the accuracy of kernel HMM compared to the KNLMS and KRLS, and its reduced level of noise in predictions.

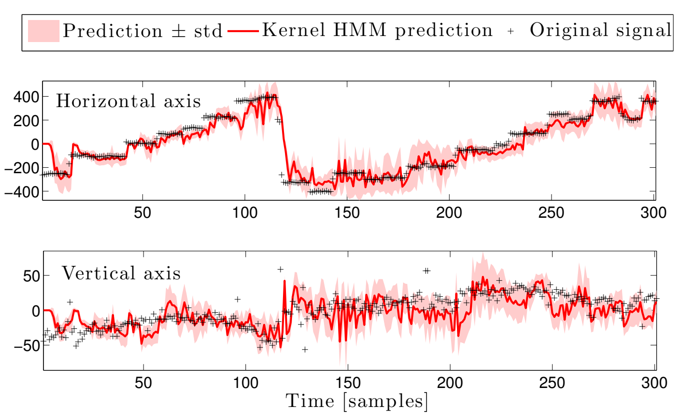

Fig. 4: Predicted mean and standard deviation within kernel HMM, evaluated for a two-dimensional point-of-gaze recording.

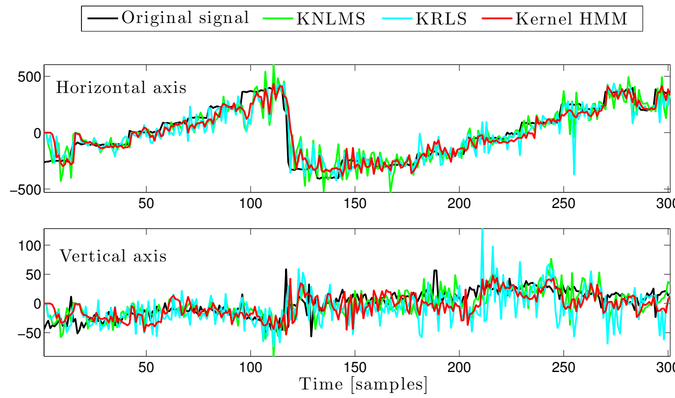

Fig. 5: Kernel algorithms prediction and original gaze signal for KNLMS, KRLS and kernel HMM.

The kernel HMM provides a full statistical description of the signal. Fig. 6 shows the prediction density and the original signal (SNR=26.02 [dB]). Observe that the prediction densities computed using kernel HMM successfully track the signal, as their probability mass is located around the original process. This makes possible for the computation of both reliable estimates and the associated confidence intervals.

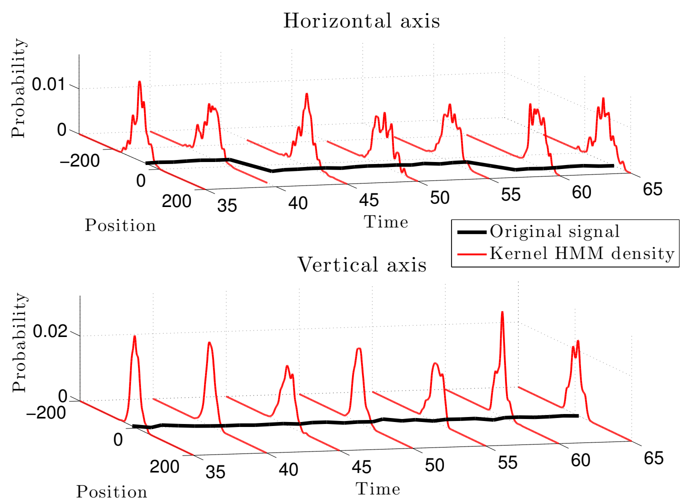

Fig. 6: Original signal and kernel HMM densities for the prediction of gaze signals.

5. CONCLUSIONS

A particle filtering based kernel HMM has been introduced for the prediction of nonlinear and nonstationary signals. The proposed algorithm models the signal as the latent state of an HMM, whereby the transition density is approximated using kernels and the weights are learnt in an unsupervised fashion using sequential importance resampling. The kernel HMM has been validated for the prediction of a synthetic nonlinear signal and real-world point-of-gaze recordings, highlighting three desired properties of the kernel HMM predictor: (i) steady state accuracy comparable to those of standard kernel estimation algorithms (KNLMS and KRLS), (ii) robustness to high observation noise, and (iii) full statistical description of its predictions, which allows for the estimation of confidence intervals.

6. REFERENCES

[1] A. Bain and D. Crisan, *Fundamentals of Stochastic Filtering*. Springer, 2009.

[2] A. H. Sayed, *Fundamentals of Adaptive Filtering*. John Wiley, New Jersey, 2003.

[3] P. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. Bugallo, and J. Miguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[4] B. Ristic, S. Arulampalm, and N. J. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Publishers, 2004.

[5] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, ser. Statistics for engineering and information science. Springer, 2001.

[6] N. Aronszajn, "Theory of reproducing kernels," *Trans. of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.

[7] A. Aizerman, E. M. Braverman, and L. I. Rozoner, "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and Remote Control*, vol. 25, pp. 821–837, 1964.

[8] H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. Advances in Neural Information Processing Systems 9*, 1996, pp. 155–161.

[9] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. on Signal Processing*, vol. 52, no. 8, pp. 2275–2285, 2004.

[10] W. Liu, P. P. Pokharel, and J. C. Principe, "The kernel least-mean-square algorithm," *IEEE Trans. on Signal Processing*, vol. 56, no. 2, pp. 543–554, 2008.

[11] W. D. Parreira, J. C. Bermudez, C. Richard, and J.-Y. Tourneret, "Stochastic behavior analysis of the Gaussian kernel least-mean-square algorithm," *IEEE Trans. on Signal Processing*, vol. 60, no. 5, pp. 2208–2222, 2012.

[12] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Trans. on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, 2009.

[13] F. Tobar, S.-Y. Kung, and D. Mandic, "Multikernel least mean square algorithm," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 265–277, 2014.

[14] F. Tobar and D. Mandic, "The quaternion kernel least squares," in *Proc. of IEEE ICASSP*, 2013, pp. 6128–6132.

[15] P. Bouboulis, S. Theodoridis, and M. Mavroforakis, "The augmented complex kernel LMS," *IEEE Trans. on Signal Processing*, vol. 60, no. 9, pp. 4962–4967, 2012.

[16] F. Tobar and D. Mandic, "Multikernel least squares estimation," in *Proc. of the Sensor Signal Processing for Defence Conference*, 2012, pp. 1–5.

[17] A. Kuh and D. P. Mandic, "Applications of complex augmented kernels to wind profile prediction," in *Proc. of IEEE ICASSP*, 2009, pp. 3581–3584.

[18] F. Tobar, A. Kuh, and D. Mandic, "A novel augmented complex valued kernel LMS," in *Proc. of the 7th IEEE Sensor Array and Multichannel Signal Processing Workshop*, 2012, pp. 473–476.

[19] G. C. Cawley, N. L. C. Talbot, R. J. Foxall, S. R. Dorling, and D. P. Mandic, "Heteroscedastic kernel ridge regression," *Neurocomputing*, vol. 57, pp. 105–124, 2004.

[20] G. Kitagawa and S. Sato, *Sequential Monte Carlo Methods in Practice*. Springer, 2001, ch. Monte Carlo Smoothing and Self-Organising State-Space Model.

[21] A. A. Georgiev, "Nonparametric system identification by kernel methods," *Automatic Control, IEEE Transactions on*, vol. 29, no. 4, pp. 356–358, 1984.

[22] L. Song, B. Boots, S. M. Siddiqi, G. J. Gordon, and A. J. Smola, "Hilbert space embeddings of hidden Markov models," in *Proc. of the 27th International Conference on Machine Learning*, 2010, pp. 991–998.

[23] P. de Valpine, "Monte Carlo state-space likelihoods by weighted posterior kernel density estimation," *Journal of the American Statistical Association*, vol. 99, pp. 523–536, 2004.

[24] C. Andrieu, N. de Freitas, and A. Doucet, "Robust full Bayesian learning for radial basis networks," *Neural Comput.*, vol. 13, no. 10, pp. 2359–2407, 2001.

[25] B. Scholkopf and A. Smola, *Learning with kernels: Support vector machines, regularization, optimization, and beyond.* Cambridge, MA, USA: MIT Press, 2001.

[26] Y. Engel, S. Mannor, and R. Meir, "Sparse online greedy support vector regression," in *13th European Conference on Machine Learning*, 2002, pp. 84–96.

[27] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 67–93, 2001.

[28] J. Liu and M. West, *Sequential Monte Carlo Methods in Practice*. Springer, 2001, ch. Combined parameter and state estimation in simulation-based filtering.

[29] F. Tobar, M. Orchard, D. Mandic, and A. Constantinides, "Estimation of financial indices volatility using a model with time-varying parameters," in *Proc. of IEEE Computational Intelligence for Financial Engineering and Economics (to appear)*, 2014.

[30] N. Gordon, D. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107–113, 1993.