The Nonparametric Kernel Bayes Smoother

Yu Nishiyama
University of Electro-Communications

Amir Hossein Afsharinejad
Georgia Tech

Shunsuke Naruse
NTT DATA Mathematical Systems Inc.

Byron Boots
Georgia Tech

Le Song
Georgia Tech

Abstract

Recently, significant progress has been made developing kernel mean expressions for Bayesian inference. An important success in this domain is the nonparametric kernel Bayes' filter (nKB-filter), which can be used for sequential inference in state space models. We expand upon this work by introducing a smoothing algorithm, the nonparametric kernel Bayes' smoother (nKB-smoother) which relies on kernel Bayesian inference through the kernel sum rule and kernel Bayes' rule. We derive the smoothing equations, analyze the computational cost, and show smoothing consistency. We summarize the algorithm, which is simple to implement, requiring only matrix multiplications and the output of the nKB-filter. Finally, we report experimental results that compare the nKB-smoother to previous parametric and nonparametric approaches to Bayesian filtering and smoothing. In the supplementary materials, we show that the combination of the nKB-filter and the nKB-smoother allows marginal kernel mean computation, which gives an alternative to kernel belief propagation.

1 Introduction

Many problems considered in machine learning and robotics, as well as the biological and natural sciences involve inferring the latent state of a dynamical system, so state space model, from sequences of observations. When uncertainty in state space models is Gaussian or multinomial, well-known algorithms for efficient inference exist. Examples include the Kalman filter and the Kalman smoother for linear Gaussian systems [15, 22] and the forward-backward algorithm for HMMs [2]. However, in many application domains, probability distributions are difficult to characterize analytically, and parametric algorithms may not be appropriate.

To combat this problem, nonparametric methods have been developed for representing and reasoning about probability distributions. Of particular interest are Hilbert space embeddings, which represent probability distributions as expectations in a reproducing kernel Hilbert space (RKHS) [24]. This embedding approach has several benefits over previous nonparametric methods: (i) closeness and similarity in the set of probability distributions is introduced via the RKHS norm, which is useful for various machine learning algorithms [12, 11, 9, 20, 30]; and (ii) estimation is relatively easy compared to nonparametric density estimation, which can be problematic when the domain is high-dimensional or structured.

In particular, the *kernel mean map* [24], defined as an expectation of feature functions that map data to an RKHS, can provide a unique embedding of a probability distribution in an RKHS. If the mapping between probability distributions and the RKHS is one-to-one, the positive definite kernel is called *characteristic* [8, 29]. It is known that typical kernels used in machine learning, for example Gaussian and Laplacian kernels, are characteristic.

Recently, significant progress has been made developing kernel mean expressions for Bayesian inference [28]. Kernelized versions of well-known probabilistic operations, have been implemented for kernel mean representations including the *kernel sum rule* (KSR) [25, 26, 27], *kernel chain rule* [25], *kernel product rule* [25], and *kernel Bayes' rule* (KBR) [10]. These methods allow *nonparametric* inference, by performing nonparametric estimation of conditional probability distributions in the supervised learning setting.[1] Since

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 41.

[1] More precisely, *conditional kernel means*, which represent conditional probability distributions in RKHS, are

they are nonparametric, we call these methods *nonparametric kernel Bayesian inference*, *nonparametric KSR (nKSR)*, *nonparametric KBR (nKBR)*, and so on, in this paper.

Various nonparametric kernel Bayesian algorithms have been developed by combining the operations. Examples include the *nonparametric kernel Bayes' filter (nKB-filter)* on discrete-time time-invariant state space models [10] and the Semiparametric *kernel Monte-Carlo filter (KMC-filter)* [16], as well as methods for dynamical system modeling and reinforcement learning applications [14, 21, 23, 4, 3, 5].

Although the nKB-filter has been developed previously for discrete-time time-invariant state space models, the corresponding smoothing algorithm was unknown. In this paper we present the *nonparametric kernel Bayes' smoother (nKB-smoother)* on state space models.[2] The nKB-smoother is applied to the output of the nKB-filter and sequentially estimates kernel means of the smoothing distributions. Like the nKB-filter, the nKB-smoother employs matrix multiplications (involving Gram matrices) to output the smoothing kernel means.

The nKB-smoother has several advantages over parametric smoothing algorithms: the nKB-smoother can be applied to any domain with hidden states and measurements, such as images, graphs, strings, and documents that are not easily modeled in a Euclidean space $\mathbb{R}^d$, provided that a similarity is defined on the domain by a positive definite kernel. And, the nKB-smoother provides kernel mean smoothing in the setting where transition and measurement distributions are both nonparametrically learned as regressions. Hence, the nKB-smoother is effective when the transition and/or measurement distributions are complicated and do not conform to simple probabilistic models.

Gaussian process (GP) regression has been used for nonparametric filtering and smoothing [17, 6, 7], and has some of the benefits of our approach. However, while GP models can work well if error has a unimodal distribution, they work much less well when error has a multimodal distribution. Previous experiments have shown that the kernel mean approach is superior than GP-based approaches if noises are multimodal [19, 18], and our experiments confirm that this is true for smoothing as well.

Although the nKB-smoother requires training data consisting of hidden states, this is inevitable for nonparametric learning of both of transition and measurement processes as regressions. The same assumption is made in the nKB-filter [10] and the GP-based filter/smoother [17, 6, 7].

Finally, in our supplementary material, we show how to extend the nKB-filter and the nKB-smoother to general tree graphs.

2 Preliminaries: Nonparametric Kernel Bayesian Inference

In this section, we introduce notation used throughout the remainder of the paper.

Positive-definite (p.d.) kernel: Let $\mathcal{X}$ be a nonempty set. A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a *positive-definite kernel* if for $\forall n \in \mathbb{N}$ and $\forall x_1, \ldots, x_n \in \mathcal{X}$, $n \times n$ matrix $G = (k(x_i, x_j))_{ij}$, $i, j \in \{1, \ldots, n\}$ is positive-semidefinite. The positive-semidefinite matrix G is called a *Gram matrix*. The function $k(\cdot, x)$ as a function of $(\cdot)$ is called the *feature function* of $x \in \mathcal{X}$.

Reproducing kernel Hilbert space (RKHS): For any positive-definite kernel k, there exists a unique reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ [Moore-Aronszajn Theorem]. The RKHS $\mathcal{H}$ associated with kernel k on a set $\mathcal{X}$ is the Hilbert space consisting of functions $f : \mathcal{X} \to \mathbb{R}$, which satisfies the following: (i) $k(\cdot, x) \in \mathcal{H}$ for any $x \in \mathcal{X}$, (ii) span $\mathcal{H}_0 := \mathrm{Span}\{k(\cdot, x) | x \in \mathcal{X}\}$ is dense in $\mathcal{H}$, i.e., $\mathcal{H} = \overline{\mathcal{H}_0}$, and (iii) there is the *reproducing property*

$$f(x) = \langle f, k(\cdot, x) \rangle_{\mathcal{H}}, \quad \forall f \in \mathcal{H}, \quad \forall x \in \mathcal{X},$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the inner product of $\mathcal{H}$. We write a triplet $(\mathcal{X}, k, \mathcal{H})$ to denote the RKHS $\mathcal{H}$ generated by a positive-definite kernel k on a domain $\mathcal{X}$.

The Kernel mean & characteristic kernel: Let $\mathcal{P}$ be the set of probability distributions on a measurable space $(\mathcal{X}, \mathcal{B}_{\mathcal{X}})$. Let $\mathbb{E}_{X \sim P}[f(X)] := \int_{\mathcal{X}} f(x) dP(x)$ be the expectation of a measurable function $f : \mathcal{X} \to \mathbb{R}$ w.r.t. a probability distribution $P \in \mathcal{P}$ of a random variable X. The *kernel mean* of $P \in \mathcal{P}$ in RKHS $\mathcal{H}$ associated with a measurable and bounded[3] p.d. kernel k is the RKHS element:

$$m_P := \mathbb{E}_{X \sim P}[k(\cdot, X)] \in \mathcal{H}. \tag{1}$$

We also use the notation m_X for random variable X. The kernel mean defines a mapping $\mathcal{P} \to \mathcal{H}; P \mapsto m_P$. If the mapping is injective (one-to-one), the p.d. kernel k is called *characteristic* [9, 29][4]. If k is characteristic, m_P uniquely specifies the probability distribution

nonparametirically estimated from a sample.

[2] This paper focuses on discrete-time time-invariant state space models. We just call them state space models.

[3] A positive-definite kernel k is bounded if $\sup_{x \in \mathcal{X}} k(x, x) < \infty$.

[4] Characteristic kernel is an analogy of the *characteristic function* $\mathbb{E}_{X \sim P}[e^{\sqrt{-1}\theta^T X}]$. Similar to the fact that a characteristic function uniquely specifies a probability dis-

P and m_P gives a representation of P in the RKHS $\mathcal{H}$. This is true even when P is complex-shaped, the RKHS function m_P is smooth, and estimation of m_P is relatively easier. *Kernel Bayesian inference* infers the representation m_P of P in $\mathcal{H}$, where P implies a predictive distribution, posterior distribution, and so on. The kernel mean m_P has the following *expectation property*:

$$\langle m_P, f\rangle_{\mathcal{H}} = \mathbb{E}_{X\sim P}[f(X)], \quad \forall f \in \mathcal{H}. \tag{2}$$

This property implies that if we have an estimator $\hat{m}_P$ of m_P, $\mathbb{E}_{X\sim P}[f(X)]$ is estimated by RKHS inner product between $\hat{m}_P$ and f. Having defined the kernel mean, we now proceed to nonparametric estimation of the kernel mean m_P.

Nonparametric kernel mean estimation: If we have n data $D_n := \{X_1, \ldots, X_n\}$ in $\mathcal{X}$, we define a finite-dimensional span $\mathcal{H}_{D_n} := \mathrm{Span}\{k(\cdot, x)|x \in D_n\}$. We estimate m_P by an element $\hat{m}_P = \sum_{i=1}^n w_i k(\cdot, X_i)$ in $\mathcal{H}_{D_n}$. If data D_n are drawn i.i.d. from P, then m_P is estimated by its sample mean $\hat{m}_P = \frac{1}{n}\sum_{i=1}^n k(\cdot, X_i)$, so that weights $w := (w_1, \ldots, w_n)^\top$ are uniform $w_i = \frac{1}{n}$. If D_n are not drawn i.i.d. from P, then the weights w are non-uniform. Weights w should be appropriately set so that the estimator $\hat{m}_P$ is consistent $||\hat{m}_P - m_P||_{\mathcal{H}} \xrightarrow{p} 0$ as $n \to \infty$. *Nonparametric kernel Bayesian inference* aims at computation of appropriate weights w expressing $\hat{m}_P$, where P implies a predictive distribution, posterior distribution, and so on. Given weights w expressing $\hat{m}_P$, the expectation of any RKHS function $f \in \mathcal{H}$ can be estimated from (2):

$$\begin{aligned}\mathbb{E}_{X\sim P}[f(X)] &\approx \langle \hat{m}_P, f\rangle_{\mathcal{H}} \\ &= \sum_{i=1}^n w_i f(\tilde{X}_i) =: \hat{\mathbb{E}}_{X\sim P}[f(X)].\end{aligned} \tag{3}$$

The Kernel sum rule (KSR) & Kernel Bayes' rule (KBR): The kernel sum rule (KSR) executes the *sum rule* in RKHSs in the kernel mean form. Kernel Bayes' rule (KBR) executes *Bayes' rule* in RKHSs in the kernel mean form. The details are as follows.

Let $(\mathcal{X}, \mathcal{B}_{\mathcal{X}})$ and $(\mathcal{Y}, \mathcal{B}_{\mathcal{Y}})$ be measurable spaces and let (X, Y) be a random variable on $\mathcal{X}\times\mathcal{Y}$ with a joint probability distribution $P_{\mathcal{X}\times\mathcal{Y}}$. Let $P_{\mathcal{X}}$ be the marginal of $P_{\mathcal{X}\times\mathcal{Y}}$ on $\mathcal{X}$ and $P_{\mathcal{Y}|x}$ be the conditional distribution given $x \in \mathcal{X}$. We write $P_{\mathcal{Y}|\mathcal{X}} := \{P_{\mathcal{Y}|x}|x \in \mathcal{X}\}$. Let Π be another probability distribution on $\mathcal{X}$ and let $Q_{\mathcal{X}\times\mathcal{Y}}$ be the joint probability distribution given by $P_{\mathcal{Y}|\mathcal{X}}$ and Π. Let $Q_{\mathcal{Y}}$ be the marginal of $Q_{\mathcal{X}\times\mathcal{Y}}$ on $\mathcal{Y}$ and let $Q_{\mathcal{X}|y}$ be the conditional distribution of $Q_{\mathcal{X}\times\mathcal{Y}}$ given $y \in \mathcal{Y}$. For ease of explanation, we assume that they have probability density functions (pdf) $p(x, y)$, $p(x)$, $p(y|x)$, $\pi(x)$, $q(x, y)$, $q(y)$, and $q(x|y)$, respectively.[5] *The sum rule* is defined by marginalization $q(y) = \int_{\mathcal{X}} p(y|x)\pi(x)dx$. *Bayes' rule* is defined by $q(x|y) = p(y|x)\pi(x)/q(y)$ with the likelihood $p(y|x)$ of observation y and prior $\pi(x)$.

Let $(\mathcal{X}, k_{\mathcal{X}}, \mathcal{H}_{\mathcal{X}})$ and $(\mathcal{Y}, k_{\mathcal{Y}}, \mathcal{H}_{\mathcal{Y}})$ be RKHSs. KSR is the computation of the marginal kernel mean $m_{Q_{\mathcal{Y}}} := \mathbb{E}_{Y\sim Q_{\mathcal{Y}}}[k_{\mathcal{Y}}(\cdot, Y)]$ given an input kernel mean $m_\Pi := \mathbb{E}_{X\sim\Pi}[k_{\mathcal{X}}(\cdot, X)]$. KBR is the computation of the posterior kernel mean $m_{Q_{\mathcal{X}|y}} := \mathbb{E}_{X\sim Q_{\mathcal{X}|y}}[k_{\mathcal{X}}(\cdot, X)]$ given y and a prior kernel mean m_Π. Nonparametric algorithms for fulfilling KSR and KBR have been proposed, which we call *nonparametric KSR (nKSR)* [25] and *nonparametric KBR (nKBR)* [10], respectively.

The nKSR algorithm is obtained by a function-valued kernel ridge regression [13]. Let $\{(X_i, Y_i)\}_{i=1}^n$ be a joint sample drawn i.i.d from $P_{\mathcal{X}\times\mathcal{Y}}$. Suppose that Π is the delta distribution $\Pi = \delta_{\tilde{X}}$ on a point $\tilde{X} \in \mathcal{X}$, i.e., $m_\Pi = k_{\mathcal{X}}(\cdot, \tilde{X})$. Then the conditional kernel mean $m_{P_{\mathcal{Y}|\tilde{X}}} (= m_{Q_{\mathcal{Y}}})$ is estimated by $\hat{m}_{P_{\mathcal{Y}|\tilde{X}}} = \sum_{i=1}^n w_i k_{\mathcal{Y}}(\cdot, Y_i)$, where $w = (w_1, \ldots, w_n)^\top$ is the kernel ridge regression weights $w = (G_X + n\epsilon_n I_n)^{-1}\mathbf{k}_{\mathcal{X}}(\tilde{X})$. Here $G_X = (k_{\mathcal{X}}(X_i, X_j))_{ij} \in \mathbb{R}^{n\times n}$ is the Gram matrix, ε_n is the regularization constant, and $\mathbf{k}_{\mathcal{X}}(\tilde{X}) := (k_{\mathcal{X}}(\tilde{X}, X_1), \ldots, k_{\mathcal{X}}(\tilde{X}, X_n))^\top \in \mathbb{R}^n$. For a general input kernel mean estimator $\hat{m}_\Pi := \sum_{i=1}^l \gamma_i k_{\mathcal{X}}(\cdot, \tilde{X}_i)$, output $\hat{m}_{Q_{\mathcal{Y}}}$ is estimated by the weighted sum, i.e., weights w are

$$w = (G_X + n\varepsilon_n I_n)^{-1} G_{X\tilde{X}}\gamma =: M^{(nKSR)}\gamma. \tag{4}$$

Here $G_{X\tilde{X}} = (k_{\mathcal{X}}(X_i, \tilde{X}_j))_{ij} \in \mathbb{R}^{n\times l}$ is the kernel matrix among data $\{X_i\}_{i=1}^n$ and $\{\tilde{X}_i\}_{i=1}^l$. Thus, the nKSR is fulfilled by multiplying the nKSR matrix $M^{(nKSR)}$ with input weights γ.

The nKBR algorithm [10] is obtained by the function-valued kernel ridge regression (nKSR) twice, i.e., regression $\mathcal{Y}$ on $\mathcal{X}$ with prior kernel mean m_Π and regression $\mathcal{X}$ on $\mathcal{Y}$ with an input observation y. Specifically, the nKBR estimates the posterior kernel mean $\hat{m}_{Q_{\mathcal{X}|y}} := \sum_{i=1}^n \tilde{w}_i k_{\mathcal{X}}(\cdot, X_i)$ given an estimator $\hat{m}_\Pi := \sum_{i=1}^l \gamma_i k_{\mathcal{X}}(\cdot, \tilde{X}_i)$ and observation $y \in \mathcal{Y}$. This is obtained by a matrix multiplication $\tilde{w} = M^{(nKBR)}(\hat{m}_\Pi)\mathbf{k}_{\mathcal{Y}}(y)$, where $\mathbf{k}_{\mathcal{Y}}(y) := (k_{\mathcal{Y}}(y, Y_1), \ldots, k_{\mathcal{Y}}(y, Y_n))^\top \in \mathbb{R}^n$ and $M^{(nKBR)}(\hat{m}_\Pi) \in \mathbb{R}^{n\times n}$ is the nKBR matrix depending on $\hat{m}_\Pi$. The nKBR matrix is given by

$$\begin{aligned}&M^{(nKBR)}(\hat{m}_\Pi) \\ &= \mathrm{diag}(w)G_Y((\mathrm{diag}(w)G_Y)^2 + \delta_n I_n)^{-1}\mathrm{diag}(w)\end{aligned} \tag{5}$$

where $\mathrm{diag}(w) \in \mathbb{R}^{n\times n}$ is the diagonal matrix of the

tribution P by the inverse Fourier transform, characteristic kernel k allows kernel mean m_P to uniquely specify a probability distribution P.

[5] The existence of pdfs is not necessary in the kernel Bayesian inference. Even if P is not absolutely continuous, m_P is a smooth RKHS function.

nKSR weights (4), G_Y is the Gram matrix of $(G_Y)_{ij} = k_{\mathcal{Y}}(Y_i, Y_j)$, and δ_n is a regularization constant.

Consistency of estimators $\hat{m}_{Q_{\mathcal{Y}}}$ and $\hat{m}_{Q_{\mathcal{X}|y}}$ when the nKSR and nKBR matrices are used, is proved in Fukumizu et al. [10], Song et al. [28].

3 Kernel Bayesian Smoothing

We propose a new addition to existing tools for nonparametric Bayesian inference: the nonparametric kernel Bayes' smoother (nKB-smoother).

3.1 Nonparametric Learning for State Space Models

The goal is to first learn the RKHS representation of the transition and measurement processes of a very general class of state space models and then to infer the hidden states. Let $\mathcal{X}$, $\mathcal{Z}$ be the domains of hidden states and observations, respectively. We consider discrete-time time-invariant models with the following properties.

- **Transition Process**: The transition of a hidden state $x \in \mathcal{X}$ to the next $x' \in \mathcal{X}$ is a general time-invariant conditional distribution $P_{\mathcal{X}'|\mathcal{X}}$.[6] Importantly, we *do not* make classic assumptions about the distribution $P_{\mathcal{X}'|\mathcal{X}}$, e.g. that it has density, is an additive noise model $x_{t+1} = f(x_t) + \varsigma_s$, or that the domain is restricted to Euclidean space $\mathbb{R}^d$. Instead, we define an RKHS $(\mathcal{X}, k_{\mathcal{X}}, \mathcal{H}_{\mathcal{X}})$ on the domain $\mathcal{X}$. Given transition data $\{(\tilde{X}_i, \tilde{X}'_i)\}_{i=1}^{l}$[7] we nonparametrically learn the conditional kernel mean of $P_{\mathcal{X}'|\mathcal{X}}$ from the data.
- **Measurement Process**: The observation of $z \in \mathcal{Z}$ given the hidden state $x \in \mathcal{X}$ is a general time-invariant conditional distribution $P_{\mathcal{Z}|\mathcal{X}}$. Again, we do not assume that $P_{\mathcal{Z}|\mathcal{X}}$ has a density, is an additive noise model $z = g(x) + \varsigma_o$, or the domain $\mathcal{X}$ and $\mathcal{Z}$ are restricted to Euclidean spaces. We again define RKHSs $(\mathcal{X}, k_{\mathcal{X}}, \mathcal{H}_{\mathcal{X}})$ and $(\mathcal{Z}, k_{\mathcal{Z}}, \mathcal{H}_{\mathcal{Z}})$ on the domains $\mathcal{X}$ and $\mathcal{Z}$, respectively. We assume data $\{(X_i, Z_i)\}_{i=1}^{n}$ and nonparametrically learn the conditional kernel mean of $P_{\mathcal{Z}|\mathcal{X}}$ from the data.

In some applications, data $\{(\tilde{X}_i, \tilde{X}'_i)\}_{i=1}^{l}$ and $\{(X_i, Z_i)\}_{i=1}^{n}$ may be obtained from a single trajectory, i.e., data have the restriction $\tilde{X}'_i = \tilde{X}_{i+1}$ $(i = 1, \ldots, l-1)$ and $\tilde{X}_i = X_i$ $(i = 1, \ldots, l-1$ and $l = n)$. However, we present the KB-smoother in the general data setting. For ease of explanation, we assume that there exist pdfs $p(x'|x)$ and $p(z|x)$ for transition and measurement processes, respectively, though, as mentioned above, this assumption is not required.

[6] We use notation $\mathcal{X}'$ for the domain of the next state. In fact, $\mathcal{X}'$=$\mathcal{X}$.

[7] We use hidden state data for nonparametric learning of $P_{\mathcal{X}'|\mathcal{X}}$ and $P_{\mathcal{Z}|\mathcal{X}}$. The same setting is put in Gaussian process approach to state space models. See also introduction about this assumption.

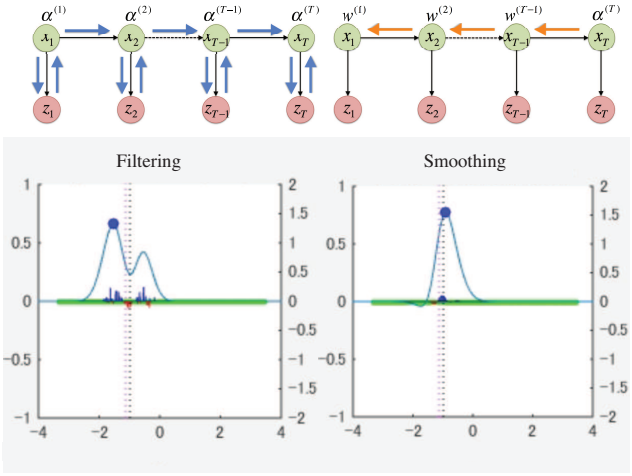

Figure 1: (Upper Left) Forward inference of the kernel mean (6) in the nKB-filter. (Upper Right) Backward inference of the kernel mean (7) in the nKB-smoother. (Lower two figures) Snapshots of filtering and smoothing results on the same variable x_t. Time-evolution of filtering and smoothing kernel means is visualized in the supplementary video (see also Fig. 4 in the supplementary material). A kernel mean (an RKHS function) is plotted as a cyan curve. Small blue and red bars indicate positive and negative weights of a kernel mean, respectively. The filtering estimation is bi-modal, but smoothing estimation correctly identifies the state by using future measurements $z_{t+1:T}$.

3.2 The Nonparametric Kernel Bayes Smoother (nKB-smoother)

The nKB-smoother is applied to the output of the nKB-filter. In general, *filtering* at time $T \geq 1$ computes the probability $p(x_T|z_{1:T})$ of the hidden state x_T given a sequence of observations $z_{1:T} := \{z_1, \ldots, z_T\}$. *Smoothing* computes the probability $p(x_t|z_{1:T})$ of past hidden state x_t $(1 \leq t < T)$ given a sequence of observations $z_{1:T}$. In other words, smoothing gives an accurate estimation of hidden state x_t $(1 \leq t < T)$ by correcting the filtering estimate using the future observations $z_{t+1:T}$.

Let kernel means of filtering and smoothing distributions be

$$m_{X_t|z_{1:t}} := \mathbb{E}_{X_t \sim p(x_t|z_{1:t})}[k_{\mathcal{X}}(\cdot, X_t)], \quad (1 \leq t \leq T),$$
$$m_{X_t|z_{1:T}} := \mathbb{E}_{X_t \sim p(x_t|z_{1:T})}[k_{\mathcal{X}}(\cdot, X_t)], \quad (1 \leq t < T).$$

We estimate the kernel means in the following non-parametric forms:

$$\hat{m}_{X_t|z_{1:t}} := \sum_{i=1}^{n} \alpha_i^{(t)} k_{\mathcal{X}}(\cdot, X_i), \quad (1 \le t \le T), \tag{6}$$

$$\hat{m}_{X_t|z_{1:T}} := \sum_{i=1}^{l} w_i^{(t)} k_X(\cdot, \tilde{X}_i), \quad (1 \le t < T). \tag{7}$$

Weights $\alpha^{(t)} := (\alpha_1^{(t)}, \ldots, \alpha_n^{(t)})$ of the filtering kernel means (6) are sequentially estimated using the nKB-filter [10]. Therefore, in this paper, we suppose that filtering weights $\{\alpha^{(t)}\}_{t=1}^{T}$ are already given.

The objective of the nKB-smoother is to estimate smoothing weights $w^{(t)} := (w_1^{(t)}, \ldots, w_l^{(t)})$. Figure 1 shows the forward and backward inference of the nKB-filter and the nkB-smoother, respectively. We derive the sequential updates for smoothing weights $\{w^{(t)}\}_{t=1}^{T-1}$.

The smoothing kernel mean (7) can be rewritten as follows:

$$\begin{aligned}
&m_{X_t|z_{1:T}} \\
&:= \int k_{\mathcal{X}}(x_t, \cdot) p(x_t|z_{1:T}) dx_t \\
&= \int k_{\mathcal{X}}(x_t, \cdot) \int p(x_t, x_{t+1}|z_{1:T}) dx_{t+1} dx_t \\
&= \int k_{\mathcal{X}}(x_t, \cdot) \int p(x_t|x_{t+1}, z_{1:t}) p(x_{t+1}|z_{1:T}) dx_{t+1} dx_t \\
&= \int \int k_{\mathcal{X}}(x_t, \cdot) p(x_t|x_{t+1}, z_{1:t}) dx_t p(x_{t+1}|z_{1:T}) dx_{t+1} \\
&= \int m_{X_t|x_{t+1}, z_{1:t}} p(x_{t+1}|z_{1:T}) dx_{t+1}, (1 \le t < T),
\end{aligned} \tag{8}$$

where $m_{X_t|x_{t+1},z_{1:t}}$ is the conditional kernel mean of $p(x_t|x_{t+1}, z_{1:t})$. Suppose that $m_{X_t|(\cdot),z_{1:t}}(x) \in \mathcal{H}_{\mathcal{X}}$ holds for $x \in \mathcal{X}$ as a function of $(\cdot)$. Then, equation (8) leads to the following backward equation for smoothing kernel means $\{m_{X_t|z_{1:T}}\}_{t=1}^{T-1}$ over time:

$$m_{X_t|z_{1:T}}(x) = \left\langle m_{X_t|(\cdot),z_{1:t}}(x), m_{X_{t+1}|z_{1:T}} \right\rangle_{\mathcal{H}_{\mathcal{X}}}, \tag{9}$$

where its initialization is the filtering kernel mean $m_{X_T|z_{1:T}}$. The conditional kernel mean $m_{X_t|x_{t+1},z_{1:t}}$ can be estimated by using the nKBR [10], where its likelihood is $p(x'|x)$ and its prior is the filtering kernel mean $m_{X_t|z_{1:t}}$, which follows from

$$\begin{aligned}
m_{X_t|x_{t+1},z_{1:t}} &= \int k(x_t, \cdot) p(x_t|x_{t+1}, z_{1:t}) dx_t \\
&= \int k(x_t, \cdot) \frac{p(x_{t+1}|x_t) p(x_t|z_{1:t})}{p(x_{t+1}|z_{1:t})} dx_t.
\end{aligned}$$

Thus, let $\hat{m}_{X_t|x_{t+1},z_{1:t}}$ be the nKBR estimator:

$$\hat{m}_{X_t|x_{t+1},z_{1:t}} = \sum_{i=1}^{l} \gamma_i^{(t,x_{t+1})} k_{\mathcal{X}}(\cdot, \tilde{X}_i). \tag{10}$$

By substituting filtering and smoothing estimators (6), (7) and the nKBR estimator (10) into the backward equation (9), we obtain the following backward equation of smoothing weights $w^{(t)}$ $(1 \le t \le T-1)$:

- When $t = T-1$,

$$\begin{aligned}
\hat{m}_{X_{T-1}|z_{1:T}}(x) &= \left\langle \hat{m}_{X_{T-1}|(\cdot),z_{1:T-1}}(x), \hat{m}_{X_T|z_{1:T}} \right\rangle_{\mathcal{H}_{\mathcal{X}}}, \\
&= \sum_{i=1}^{n} \alpha_i^{(T)} \hat{m}_{X_{T-1}|X_i,z_{1:T-1}}(x) \\
&= \sum_{j=1}^{l} \left(\sum_{i=1}^{n} \alpha_i^{(T)} \gamma_j^{(T-1,X_i)} \right) k_{\mathcal{X}}(x, \tilde{X}_j) \\
&=: \sum_{j=1}^{l} w_j^{(T-1)} k_{\mathcal{X}}(x, \tilde{X}_j).
\end{aligned} \tag{11}$$

  From (11), the weight update is

$$w_j^{(T-1)} = \sum_{i=1}^{n} \alpha_i^{(T)} \gamma_j^{(T-1,X_i)}, \quad j = 1, \ldots, l. \tag{12}$$

- Similarly, when $1 \le t \le T-2$,

$$\begin{aligned}
\hat{m}_{X_t|z_{1:T}}(x) &= \left\langle m_{X_t|(\cdot),z_{1:t}}(x), m_{X_{t+1}|z_{1:T}} \right\rangle_{\mathcal{H}_{\mathcal{X}}} \\
&= \sum_{i=1}^{l} w_i^{(t+1)} \hat{m}_{X_t|\tilde{X}_i,z_{1:t}}(x) \\
&= \sum_{j=1}^{l} \left(\sum_{i=1}^{l} w_i^{(t+1)} \gamma_j^{(t,\tilde{X}_i)} \right) k_{\mathcal{X}}(x, \tilde{X}_j) \\
&=: \sum_{j=1}^{l} w_j^{(t)} k_{\mathcal{X}}(x, \tilde{X}_j).
\end{aligned} \tag{13}$$

  From (13), the weight update is

$$w_j^{(t)} = \sum_{i=1}^{l} w_i^{(t+1)} \gamma_j^{(t,\tilde{X}_i)}, \quad j = 1, \ldots, l. \tag{14}$$

Thus, from updates (12) and (14), we have the following **smoothing weight recursion** in matrix form:

$$\begin{aligned}
w^{(T-1)} &= \Gamma^{(T-1)} \alpha^{(T)}, \\
w^{(t)} &= \Gamma^{(t)} w^{(t+1)}, \quad (1 \le t \le T-2),
\end{aligned} \tag{15}$$

where

$$\begin{aligned}
\Gamma^{(T-1)} &:= (\gamma^{(T-1,X_1)}, \ldots, \gamma^{(T-1,X_n)}) \in \mathbb{R}^{l \times n}, \\
\Gamma^{(t)} &:= (\gamma^{(t,\tilde{X}_1)}, \ldots, \gamma^{(t,\tilde{X}_l)}) \in \mathbb{R}^{l \times l}.
\end{aligned}$$

The matrix $\Gamma^{(T-1)}$ is the collection of posterior weights of (10) at time $t = T-1$ given measurement sample $x_T = X_1, \ldots, X_n$. The matrix $\Gamma^{(t)}$ is the collection of posterior weights of (10) at time t given transition sample $x_{t+1} = \tilde{X}_1, \ldots, \tilde{X}_l$.[8] For notational simplicity, we use $w^{(T)} := \alpha^{(T)}$. Thus, smoothing weights

[8] If data $\{(\tilde{X}_i, \tilde{X}_i')\}_{i=1}^{l}$ and $\{(X_i, Z_i)\}_{i=1}^{n}$ are a single trajectory data as noted in Section 3.1, then matrix $\Gamma^{(t)}$ $(1 \le t \le T-1)$ becomes the $n \times n$ matrix $\Gamma^{(t)} :=$

$\{w^{(t)}\}_{t=1}^{T-1}$ can be computed by matrix multiplications; $w^{(t)} = (\prod_{i=T-1}^{t} \Gamma^{(i)}) w^{(T)}$, $(1 \leq t \leq T-1)$, where $\prod_{i=T-1}^{t} \Gamma^{(i)} := \Gamma^{(t)} \cdots \Gamma^{(T-1)}$.

The detailed Gram matrix expression of $\Gamma^{(t)}$ $(1 \leq t \leq T-1)$ is as follows. Let $G_{\tilde{X}}$ and $G_{\tilde{X}'}$ be Gram matrices such that $(G_{\tilde{X}})_{ij} = k_{\mathcal{X}}(\tilde{X}_i, \tilde{X}_j)$ and $(G_{\tilde{X}'})_{ij} = k_{\mathcal{X}}(\tilde{X}'_i, \tilde{X}'_j)$. The posterior weights in equation (10) is computed as

$$\gamma^{(t,x_{t+1})} = M^{(nKBR)}_{\mathcal{X}|\mathcal{X}'}(\hat{m}_{X_t|z_{1:t}}) \mathbf{k}_{\mathcal{X}'}(x_{t+1}),$$

where $M^{(nKBR)}_{\mathcal{X}|\mathcal{X}'}(\hat{m}_{X_t|z_{1:t}})$ is the nKBR matrix operating from $\mathcal{X}'$ to $\mathcal{X}$ given the transition data $\{(\tilde{X}_i, \tilde{X}'_i)\}_{i=1}^{l}$ and filtering prior $\hat{m}_{X_t|z_{1:t}}$. We note that the same matrix $M^{(nKBR)}_{\mathcal{X}|\mathcal{X}'}(\hat{m}_{X_t|z_{1:t}})$ can be used for computing $\gamma^{(t,x)}$ with different values $x = \tilde{X}_1, \ldots, \tilde{X}_l$. We can use the nKBR matrix (5) for the consistent estimator. Finally, we have the following:

- When $t = T-1$,

$$\begin{aligned}\Gamma^{(T-1)} &= (\gamma^{(T-1,X_1)}, \ldots, \gamma^{(T-1,X_n)}) \\ &= M^{(nKBR)}_{\mathcal{X}|\mathcal{X}'}(\hat{m}_{X_{T-1}|z_{1:T-1}}) G_{\tilde{X}'X} \\ &= \mathrm{diag}(\xi_{T-1}) G_{\tilde{X}'}((\mathrm{diag}(\xi_{T-1}) G_{\tilde{X}'})^2 + \tilde{\delta}_n I_n)^{-1} \\ &\quad \times \mathrm{diag}(\xi_{T-1}) G_{\tilde{X}'X}.\end{aligned}$$

- Similarly, when $1 \leq t \leq T-2$,

$$\begin{aligned}\Gamma^{(t)} &= (\gamma^{(t,\tilde{X}_1)}, \ldots, \gamma^{(t,\tilde{X}_l)}) \\ &= M^{(nKBR)}_{\mathcal{X}|\mathcal{X}'}(\hat{m}_{X_t|z_{1:t}}) G_{\tilde{X}'\tilde{X}} \\ &= \mathrm{diag}(\xi_t) G_{\tilde{X}'}((\mathrm{diag}(\xi_t) G_{\tilde{X}'})^2 + \tilde{\delta}_n I_n)^{-1} \\ &\quad \times \mathrm{diag}(\xi_t) G_{\tilde{X}'\tilde{X}}.\end{aligned}$$

Here, ξ_t are the nKSR weights:

$$\xi_t = (G_{\tilde{X}} + l\tilde{\varepsilon}_l I_l)^{-1} \hat{m}_{X_t|z_{1:t}} = (G_{\tilde{X}} + l\tilde{\varepsilon}_l I_l)^{-1} G_{\tilde{X}X} \alpha^{(t)}$$

and, $\tilde{\delta}_n$ and $\tilde{\epsilon}_l$ are their regularization constants.

Algorithm 1 summarizes the nKB-smoother.[9] Figure 1 shows an example of computed kernel means with weights $\alpha^{(t)}$ and $w^{(t)}$. Smoothing weights $w^{(t)}$ are used to compute expectations (3) of RKHS functions $\forall f \in \mathcal{H}_{\mathcal{X}}$ over smoothing distribution $p(x_t|z_{1:T})$ and the mode estimation. The mode is estimated by solving an optimization problem $\hat{x} = \arg\min_{x \in \mathcal{X}} ||\hat{m}_P - k_{\mathcal{X}}(\cdot, x)||^2 = \arg\max_{x \in \mathcal{X}} \hat{m}_P(x)$ given a smoothing estimator $\hat{m}_P \in \mathcal{H}_{\mathcal{X}}$ [10]. In experiments (Section 4), we report the mode estimation results on the nKB-smoother.

$(\gamma^{(t,X_1)}, \ldots, \gamma^{(t,X_n)})$ $(1 \leq t \leq T-1)$.

[9] Algorithm 1 is so generic that any consistent nKBR matrix is allowed over the currently used Tikhonov-type nKBR matrix [10].

Algorithm 1: The nKB-smoother

Input: filtering weights $\alpha^{(1)}, \ldots, \alpha^{(T)}$ and regularization constants $\tilde{\epsilon}_l$, $\tilde{\delta}_n$

Initialize: $\Gamma^{(T-1)} = M^{(nKBR)}_{\mathcal{X}|\mathcal{X}'}(\hat{m}_{X_{T-1}|z_{1:T-1}}) G_{\tilde{X}'X}$
$w^{(T-1)} = \Gamma^{(T-1)} \alpha^{(T)}$

for $t = T-2$ *to* $t = 1$ **do**
  $\Gamma^{(t)} = M^{(nKBR)}_{\mathcal{X}|\mathcal{X}'}(\hat{m}_{X_t|z_{1:t}}) G_{\tilde{X}'\tilde{X}}$
  $w^{(t)} = \Gamma^{(t)} w^{(t+1)}$
end

Return: smoothing weights $w^{(1)}, \ldots, w^{(T-1)}$

3.3 Computational Cost

Here we discuss the smoothing cost for a typical case $n = l$. Kernel Bayes filtering costs $O(n^3)$ for a single step of filtering because the nKBR needs a Gram matrix inversion [10]. Similarly, kernel Bayes smoothing costs $O(l^3)$ for a step of smoothing, since computation of $\Gamma^{(t)}$ costs $O(l^3)$ and its multiplication $w^{(t)} = \Gamma^{(t)} w^{(t+1)}$ costs $O(l^2)$. Thus smoothing has the same order as the filtering. Filtering and smoothing for a length T test sequence costs $O(Tn^3)$. The cost can be reduced to $O(Tnr^2)$ by using low rank approximations of rank r, as used in Fukumizu et al. [10]. In addition, we note that the KB-smoother can be made more efficient through parallel computation of the matrices $\{\Gamma^{(t)}\}_{t=1}^{T-1}$, since matrix $\Gamma^{(t)}$ does not depend on matrix $\Gamma^{(t+1)}$.

3.4 Consistency of Smoothing

We have the following consistency result for the smoothing estimator. The proof is in Appendix 6.1.

Theorem 3.1. *Let* $\xi_t(x_{t+1}, \tilde{x}_{t+1}) := \langle m_{X_t|x_{t+1},z_{1:t}}, m_{X_t|\tilde{x}_{t+1},z_{1:t}} \rangle_{\mathcal{H}_{\mathcal{X}}}$, $\Delta m_{X_t|x_{t+1},z_{1:t}} := m_{X_t|x_{t+1},z_{1:t}} - \hat{m}_{X_t|x_{t+1},z_{1:t}}$, *and* $\Delta\xi_t(x_{t+1}, \tilde{x}_{t+1}) := \langle \Delta m_{X_t|x_{t+1},z_{1:t}}, \Delta m_{X_t|\tilde{x}_{t+1},z_{1:t}} \rangle_{\mathcal{H}_{\mathcal{X}}}$. *Assume that* $\xi_t \in \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}$. *If* $\left\| m_{X_{t+1}|z_{1:T}} - \hat{m}_{X_{t+1}|z_{1:T}} \right\|_{\mathcal{H}_{\mathcal{X}}} = O_p(l^{-\alpha_{t+1}})$ *for* $\alpha_{t+1} \in (0, \frac{1}{2}]$ *and* $\|\Delta\xi_t\|_{\mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}} = O_p(l^{-2\beta_t})$ *for* $\beta_t \in (0, \frac{1}{2}]$, *then*

$$\left\| m_{X_t|z_{1:T}} - \hat{m}_{X_t|z_{1:T}} \right\|_{\mathcal{H}_{\mathcal{X}}} = O_p(l^{-\alpha_t}),$$

where $\alpha_t = \min\{\alpha_{t+1}, \beta_t\}$.

From the consistecy of filtering, it is known [10] that $\left\| m_{X_T|z_{1:T}} - \hat{m}_{X_T|z_{1:T}} \right\|_{\mathcal{H}_{\mathcal{X}}} = O_p(l^{-\alpha_T})$ for $\alpha_T \in (0, \frac{1}{2}]$. By induction with Theorem 3.1, we have the consistency of smoothing $\left\| m_{X_t|z_{1:T}} - \hat{m}_{X_t|z_{1:T}} \right\|_{\mathcal{H}_{\mathcal{X}}} = O_p(l^{-\alpha_t})$ for any $1 \leq t < T$.

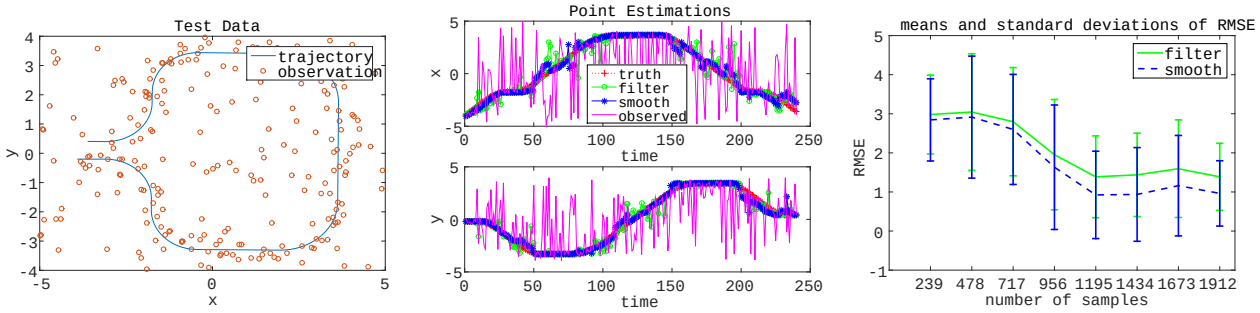

Figure 2: Smoothing in a cluttered environment. (Left) The blue curve shows the true trajectory of the object and red points the measurements. (Middle) Filtering (green) and smoothing (blue) as outputs of the nKB-filter and the nKB-smoother, respectively. The red and magenta show the true target's position $\{(x_t, y_t)\}_t$ and measurements $\{(\tilde{x}_t, \tilde{y}_t)\}_t$, respectively. More results on different training and test data are presented in the supplementary material. (Right) Averaged results of RMSEs $\{(1/240 \sum_{t=1}^{240} ||(x_t^{(i)}, y_t^{(i)}) - (\hat{x}_t^{(i)}, \hat{y}_t^{(i)})||^2)^{1/2}\}_{i=1}^{M}$ (vertical) on each training sample size n (horizontal) over $M = 10$ experiments.

4 Experimental Results

We implemented and validated the nKB-smoother on a synthetic and a real-world dataset.

4.1 Synthetic Experiment: Smoothing in a Cluttered Environment

In the first experiment we applied the nKB-filter [10] and the nKB-smoother (Algorithm 1) to the problem of "Tracking a Single Object with Cluttered Measurements," from the RBMCDA toolbox.[10]

The objective of this problem is to estimate the trajectory of a single moving object in $\mathbb{R}^2$, using measurements corrupted by clutter. The transition dynamics of the object are modeled with a standard discretized Wiener velocity model (e.g., [1]). The target's state at time t is described by $\mathbf{x}_t = (x_t, y_t, \dot{x}_t, \dot{y}_t)$ with the object's position (x_t, y_t) and the velocity $(\dot{x}_t, \dot{y}_t)$ in cartesian coordinates $\mathbb{R}^2$. The measurement for the target is w.r.t. position $\mathbf{z}_t = (\tilde{x}_t, \tilde{y}_t)$ with a mixture noise of a Gaussian and the uniform clutter. Figure 2 (left) shows the target's trajectory (blue) and measurements (red points). We assume that the transition and measurement processes are not known; instead we learn them from samples of transition pairs $\{(\mathbf{x}_i, \mathbf{x}'_i)\}_{i=1}^n$ and state-observation pairs $\{(\mathbf{x}_i, \mathbf{z}_i)\}_{i=1}^n$.

Figure 2 (middle) shows point estimation of the target's position (x_t, y_t) by the nKB-filter and the nKB-smoother. Figure 2 (right) shows the averaged performance of the nKB-filter in root mean squared error (RMSE) against training sample size n. The nKB-smoother provides a more accurate estimate of the target's position relative to the nKB-filter by using future observations $\mathbf{z}_{t+1:T}$ in the kernel mean form.

[10] Matlab codes and documents are in `http://becs.aalto.fi/en/research/bayes/rbmcda/`. For details, see the URL and documents within.

4.2 Real-World Experiment: Slotcar State Estimation

The second experiment focuses on estimating the progress of a miniature car (1:32 scale) racing around a 14m track (a similar dataset was used in Song et al. [26], Boots et al. [3]). Figure 3 shows the car and an attached 6-axis IMU (an Intel Inertiadot), as well as the track. The observations are noisy estimates of 3D acceleration and velocity of the car, collected at 10Hz from the IMU. Ground truth positional information is captured by an overhead camera that uses a particle filter to track the position of the car in the image. Despite the complexity of the track (it has several sharp U-turns and bridges), the position of the car on each lap can be described by a circular manifold. We consider the 2-D space that contains this manifold to be the latent state space.

The goal of the experiment is to infer the progress of the car on the manifold from *just* the noisy IMU data. Our training data consists of the ground-truth tracking information converted to the 2-D position on the circular manifold and aligned with IMU data. Specifically we train each algorithm on transition pairs $\{(\mathbf{x}_i, \mathbf{x}'_i)\}_{i=1}^n$ and state-observation pairs $\{(\mathbf{x}_i, \mathbf{z}_i)\}_{i=1}^n$.

To test our data, we learn linear-Gaussian models, Gaussian process models, and nonparametric models of the transition and measurement processes. We then execute filtering and smoothing to infer the progress of the car from noisy observations on a heldout test set. Performance is compared to the Kalman filter and Kalman smoother, with parameters estimated by linear regression, as well as Gaussian process extended Kalman filter (GP-EKF) and Gaussian process ex-

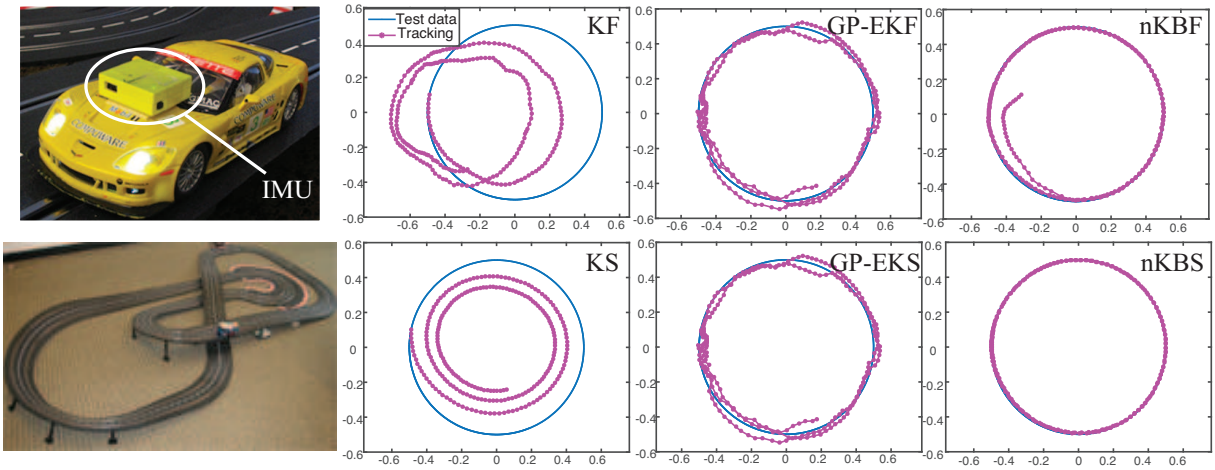

Figure 3: (Top left) The slotcar and inertial measurement unit. (Lower left) The 14m racetrack. The six graphs illustrate filtering and smoothing performance. The blue points show the ground-truth position of the car on the circular manifold representing a single lap around the track. We chose the diameter of the manifold to be one. The magenta points show the results of filtering and smoothing. For visual clarity we only plot the first 250 data ponts, but we summarize the quantitative results in Table 1. (Left) Kalman-filter and Kalman smoothing. (Middle) GP-EKF and GP-EKS (Right) nKB-filter and nKB-smoothing. The results indicate that the nKB-filter and the nKB-smoother are better able to infer the position of the car on the circular manifold compared to the linear Kalman and and Gaussian Process-based approaches.

Table 1: Filtering and Smoothing Error

Algorithm	**MSE on x_1**	**MSE on x_2**
Kalman Filter	0.1082	0.0143
Kalman Smoothing	0.0121	0.0096
GP-EKF	1.64128 e-03	1.8133 e-03
GP-EKS	1.64122 e-03	1.8132 e-03
nKB-filter	4.3901 e-04	3.1178 e-04
nKB-smoothing	**2.2087e-04**	**1.9457 e-04**

tended Kalman smoothing (GP-EKS) [17].

The training dataset consist of 7000 transition and state observation pairs and the test dataset consists of 1400 observations. In our model, the regularization parameter is set equal to 0.01 and the bandwidths of the Gaussian RBF kernels are set via cross validation to minimize the prediction error on the training data.

In Table 1, the Mean Squared Error (MSE) for prediction on the manifold with various approaches is reported. Figure 3 illustrates the performance of each algorithm inferring the latent state on the test set. nKB-smoothing improves on the nKB-filtering result and also has the highest accuracy compared to the linear Kalman and and Gaussian Process-based approaches.

5 Conclusion

We have introduced a novel algorithm for smoothing in state space models using kernel mean embeddings called the nonparametric kernel Bayes' smoother (nKB-smoother). The nKB-smoother is very general: it can be applied to any state space model with hidden states and measurements, and does not require simple probabilistic models for the transition or measurement distributions. We derive the smoothing equations, analyze the computational cost, and show smoothing consistency. Finally, we implement the nKB-smoother and show that it can achieve accurate results in practice. We believe that this paper provides an important tool for inference in state space models and that it will assist in developing more advanced algorithms, including expectation-maximization-like algorithms for semi-parametric kernel Bayesian inference.

Acknowledgements

We thank anonymous reviewers for helpful comments. Y.N. was supported by JSPS KAKENHI Grant Number 26870821 and Program to Disseminate Tenure Tracking System, MEXT, Japan. B.B. was supported in part by NSF CRII 1464219. L.S. was supported in part by NSF/NIH BIGDATA 1R01GM108341, ONR N00014-15-1-2340, NSF IIS-1218749, NSF CAREER IIS-1350983.

References

[1] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc. cop., 2001. ISBN 0-471-41655-X. URL `http://opac.inria.fr/record=b1132468`.

[2] John Binder, Kevin Murphy, and Stuart Russell. Space-efficient inference in dynamic probabilistic networks. In *Proceedings of the Fifteenth International*

Joint Conference on Artifical Intelligence - Volume 2, IJCAI'97, pages 1292–1296, 1997.

[3] B. Boots, G. Gordon, and Arthur Gretton. Hilbert Space Embeddings of Predictive State Representations. In *UAI*, pages 92–101, 2013.

[4] Byron Boots. *Spectral Approaches to Learning Predictive Representations*. PhD thesis, Carnegie Mellon University, December 2012.

[5] Byron Boots, Arunkumar Byravan, and Dieter Fox. Learning predictive models of a depth camera & manipulator from raw execution traces. In *Proceedings of The International Conference in Robotics and Automation (ICRA-2014)*, 2014.

[6] M. P. Deisenroth, M. F. Huber, and U. D. Hanebeck. Analytic Moment-based Gaussian Process Filtering. In *ICML*, pages 225–232, 2009.

[7] M.P. Deisenroth, R. Turner, M. Huber, U.D. Hanebeck, and C.E Rasmussen. Robust Filtering and Smoothing with Gaussian Processes. *IEEE Transactions on Automatic Control*, 2012.

[8] K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality Reduction for Supervised Learning with Reproducing Kernel Hilbert Spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.

[9] K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel Measures of Conditional Dependence. In *NIPS*, pages 489–496. 2008.

[10] K. Fukumizu, L. Song, and A. Gretton. Kernel bayes' rule: Bayesian inference with positive definite kernels. *Journal of Machine Learning Research*, pages 3753–3783, 2013.

[11] A. Gretton and L. Györfi. Consistent Nonparametric Tests of Independence. *Journal of Machine Learning Research*, 11:1391–1423, 2010.

[12] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13:723–773, 2012.

[13] S. Grünewälder, G. Lever, L. Baldassarre, S. Patterson, A. Gretton, and M. Pontil. Conditional mean embeddings as regressors. In *ICML*, pages 1823–1830, 2012.

[14] S. Grünewälder, G. Lever, L. Baldassarre, M. Pontil, and A. Gretton. Modelling transition dynamics in MDPs with RKHS embeddings. In *ICML*, pages 535–542, 2012.

[15] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D): 35–45, 1960.

[16] M. Kanagawa, Y. Nishiyama, A. Gretton, and K. Fukumizu. Monte Carlo Filtering Using Kernel Embedding of Distributions. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1897–1903, 2014.

[17] J. Ko and D. Fox. GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models. *Auton. Robots*, 27(1):75–90, 2009.

[18] L. McCalman. Function Embeddings for Multi-modal Bayesian Inference. *A Ph.D. thesis in the University of Sydney*, 2013 (http://hdl.handle.net/2123/12031). URL `http://hdl.handle.net/2123/12031`.

[19] L. McCalman, S. O'Callaghan, and F. Ramos. Multi-modal estimation with kernel embeddings for learning motion models. In *IEEE International Conference on Robots and Automation (ICRA)*, 2013.

[20] K. Muandet, K. Fukumizu, F. Dinuzzo, and B. Schölkopf. Learning from Distributions via Support Measure Machines. In *NIPS*, pages 10–18. 2012.

[21] Y. Nishiyama, A. Boularias, A. Gretton, and K. Fukumizu. Hilbert Space Embeddings of POMDPs. In *UAI*, pages 644–653, 2012.

[22] H.E. Rauch. Solutions to the smoothing problem. *IEEE Trans. Aut. Contr.*, Ac-8:371?372, 1963.

[23] K. Rawlik, M. Toussaint, and S. Vijayakumar. Path Integral Control by Reproducing Kernel Hilbert Space Embedding. *Proc. 23rd Int. Joint Conference on Artificial Intelligence (IJCAI)*, 2013.

[24] A. Smola, A. Gretton, L. Song, and B. Schölkopf. A Hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory (ALT)*, pages 13–31, 2007.

[25] L. Song, J. Huang, A. Smola, and K. Fukumizu. Hilbert Space Embeddings of Conditional Distributions with Applications to Dynamical Systems. In *ICML*, pages 961–968, 2009.

[26] L. Song, B. Boots, S. M. Siddiqi, G. J. Gordon, and A. J. Smola. Hilbert Space Embeddings of Hidden Markov Models. In *ICML*, pages 991–998, 2010.

[27] L. Song, A. Gretton, D. Bickson, Y. Low, and C. Guestrin. Kernel Belief Propagation. *Journal of Machine Learning Research - Proceedings Track*, 15: 707–715, 2011.

[28] L. Song, K. Fukumizu, and A. Gretton. Kernel embedding of conditional distributions. *IEEE Signal Processing Magazine*, 30(4):98–111, 2013.

[29] B. Sriperumbudur, A. Gretton, K. Fukumizu, G. Lanckriet, and B. Schölkopf. Hilbert Space Embeddings and Metrics on Probability Measures. *Journal of Machine Learning Research*, 11:1517–1561, 2010.

[30] Y. Yoshikawa, T. Iwata, and H. Sawada. Latent Support Measure Machines for Bag-of-Words Data Classification. In *Advances in Neural Information Processing Systems 27*, pages 1961–1969. 2014.

6 Supplementary Materials

6.1 Proof of Theorem 3.1

Proof. Let $\tilde{m}_{X_t|z_{1:T}} := \sum_{i=1}^{l} w_i^{(t+1)} m_{X_t|\tilde{X}_i,z_{1:t}}$. We then have

$$\left\|m_{X_t|z_{1:T}} - \hat{m}_{X_t|z_{1:T}}\right\|_{\mathcal{H}_\mathcal{X}} \leq \left\|m_{X_t|z_{1:T}} - \tilde{m}_{X_t|z_{1:T}}\right\|_{\mathcal{H}_\mathcal{X}} + \left\|\tilde{m}_{X_t|z_{1:T}} - \hat{m}_{X_t|z_{1:T}}\right\|_{\mathcal{H}_\mathcal{X}}. \tag{16}$$

We consider each of the two terms in equation (16). Let $\Delta m_{X_t|z_{1:T}} := m_{X_t|z_{1:T}} - \tilde{m}_{X_t|z_{1:T}}$. For the first term,

$$\begin{aligned}
& \left\|\Delta m_{X_t|z_{1:T}}\right\|^2_{\mathcal{H}_\mathcal{X}} \\
= & \left\|m_{X_t|z_{1:T}} - \sum_{i=1}^{l} w_i^{(t+1)} m_{X_t|\tilde{X}_i,z_{1:t}}\right\|^2_{\mathcal{H}_\mathcal{X}} \\
= & \sum_{i,j=1}^{l} w_i^{(t+1)} w_j^{(t+1)} \xi_t(\tilde{X}_i, \tilde{X}_j) \\
& -2\sum_{i=1}^{l} w_i^{(t+1)} \int \xi_t(\tilde{X}_i, x) dP_{X_{t+1}|z_{1:T}}(x) \\
& + \int \xi_t(x, \tilde{x}) dP_{X_{t+1}|z_{1:T}}(x) dP_{X_{t+1}|z_{1:T}}(\tilde{x}) \\
= & \left\langle \Delta m_{X_{t+1}|z_{1:T}} \otimes \Delta m_{X_{t+1}|z_{1:T}}, \xi_t \right\rangle_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}} \\
\leq & \left\|\Delta m_{X_{t+1}|z_{1:T}}\right\|^2_{\mathcal{H}_\mathcal{X}} \|\xi_t\|_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}}
\end{aligned}$$

Since $\|\xi_t\|_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}} < \infty$, the first term decays with $O_p(l^{-2\alpha_{t+1}})$. For the second term, we have

$$\begin{aligned}
& \left\|\tilde{m}_{X_t|z_{1:T}} - \hat{m}_{X_t|z_{1:T}}\right\|^2_{\mathcal{H}_\mathcal{X}} \\
= & \left\|\sum_{i=1}^{l} w_i^{(t+1)} (m_{X_t|\tilde{X}_i,z_{1:t}} - \hat{m}_{X_t|\tilde{X}_i,z_{1:t}})\right\|^2_{\mathcal{H}_\mathcal{X}} \\
= & \left\|\sum_{i=1}^{l} w_i^{(t+1)} \Delta m_{X_t|\tilde{X}_i,z_{1:t}}\right\|^2_{\mathcal{H}_\mathcal{X}} \\
= & \sum_{i,j=1}^{l} w_i^{(t+1)} w_j^{(t+1)} \Delta\xi_t(\tilde{X}_i, \tilde{X}_j) \\
= & \left\langle \hat{m}_{X_{t+1}|z_{1:T}} \otimes \hat{m}_{X_{t+1}|z_{1:T}}, \Delta\xi_t \right\rangle_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}} \\
\leq & \left\|\hat{m}_{X_{t+1}|z_{1:T}}\right\|^2_{\mathcal{H}_\mathcal{X}} \|\Delta\xi_t\|_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}}.
\end{aligned}$$

Since $\|\hat{m}_{X_{t+1}|z_{1:T}}\|_{\mathcal{H}_\mathcal{X}} \to \|m_{X_{t+1}|z_{1:T}}\|_{\mathcal{H}_\mathcal{X}} < \infty$, the second term decays with $O_p(l^{-2\beta_t})$. These results lead to the statement $\left\|m_{X_t|z_{1:T}} - \hat{m}_{X_t|z_{1:T}}\right\|_{\mathcal{H}_\mathcal{X}} = O_p(l^{-\alpha_t})$, where $\alpha_t = \min\{\alpha_{t+1}, \beta_t\}$. □

6.2 Experimental Setting & Video: Tracking a Single Object (Experiment 1)

State Space Model Setting: The target's state at time t is described by $\mathbf{x}_t = (x_t, y_t, \dot{x}_t, \dot{y}_t)$ with the object's position (x_t, y_t) and the velocity $(\dot{x}_t, \dot{y}_t)$ in cartesian coordinates $\mathbb{R}^2$. The discretized dynamics is expressed with

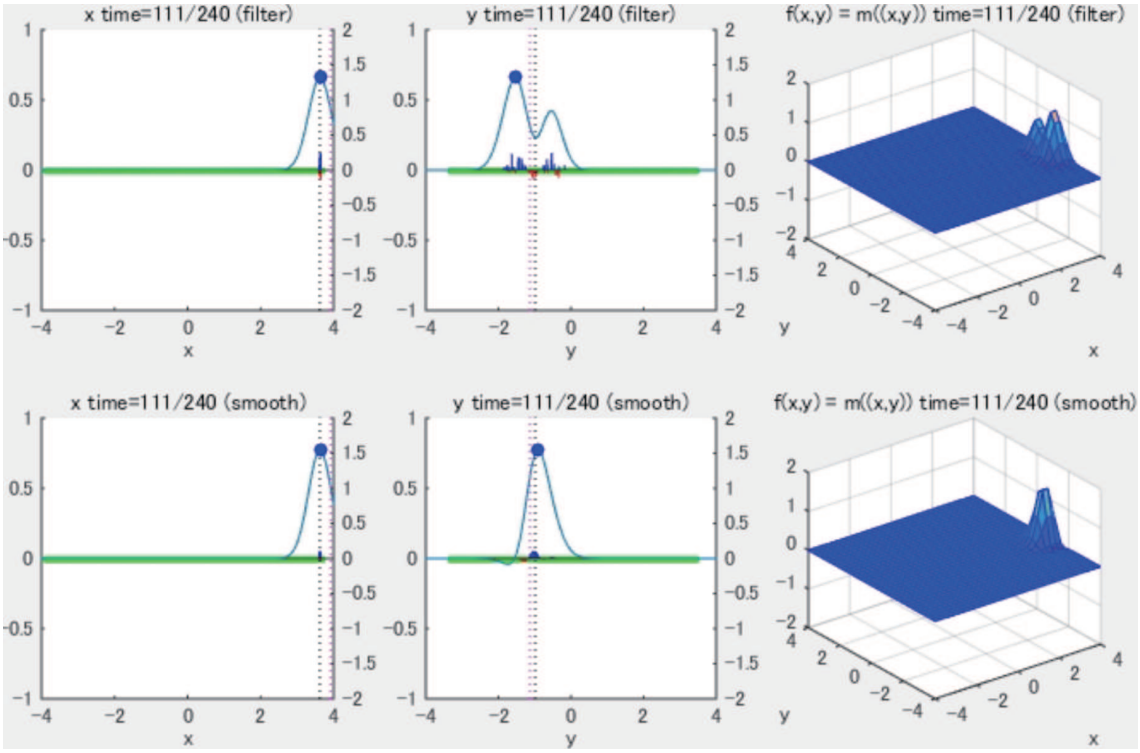

Figure 4: A supplementary video. This animation visualizes the sequential update of kernel means of the nKB-filter [10] and the nKB-smoother (Algorithm 1) for a test sequence $z_{1:240}$ in the clutter problem. The upper three figures show the sequential update of kernel means $m_{X_t|z_{1:t}}$ $(t = 1 : 240)$ of the nKB-filter. The lower three figures show the estimated smoothing kernel means $m_{X_t|z_{1:240}}$ $(t = 1 : 240)$ of the nKB-smoother. For each, the left figure shows the kernel mean projected to state x, the middle figure shows the kernel mean projected to state y, and the right figure both. Each figure visualizes the following. (Left four figures) The black dot vertical line shows the true target's state (x, y). The magenta dot vertical line shows the (cluttered) observation $(\tilde{x}, \tilde{y})$. The kernel mean weights are shown with left vertical axis. The positive (negative) weight values are visualized with blue (red) bars, respectively. The cyan curve shows the estimated kernel mean (estimated RKHS function) $m_P(\cdot) \in \mathcal{H}_\mathcal{X}$ as a function of $(\cdot)$ with right vertical axis. The blue dot in the top of the mountain shows the result of the mode estimation for the target's state (x, y) with the objective function value. From the two middle figures, it can be observed that the filtering estimation is bimodal for uncertainty, but smoothing estimation correctly identifies the state by using the future measurements $z_{112:240}$, so that the blue dot is on the black dot vertical line.

a time-invariant linear equation:

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + \mathbf{q}_t, \quad A := \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{17}$$

where $\mathbf{q}_t$ is discrete Gaussian white process noise having moments

$$\mathbb{E}[\mathbf{q}_t] = \mathbf{0},$$
$$\mathbb{E}[\mathbf{q}_t\mathbf{q}_t^\top] := \begin{pmatrix} \Delta t^3/3 & 0 & \Delta t^2/2 & 0 \\ 0 & \Delta t^3/3 & 0 & \Delta t^2/2 \\ \Delta t^2/2 & 0 & \Delta t & 0 \\ 0 & \Delta t^2/2 & 0 & \Delta t \end{pmatrix} q$$

with $q > 0$. The measurement process for the target is a mixture model:

$$p(\mathbf{z}_t|\mathbf{x}_t) = (1 - \rho)N(\mathbf{z}_t|H\mathbf{x}_t, R) + \rho\frac{1}{|S|}, \tag{18}$$

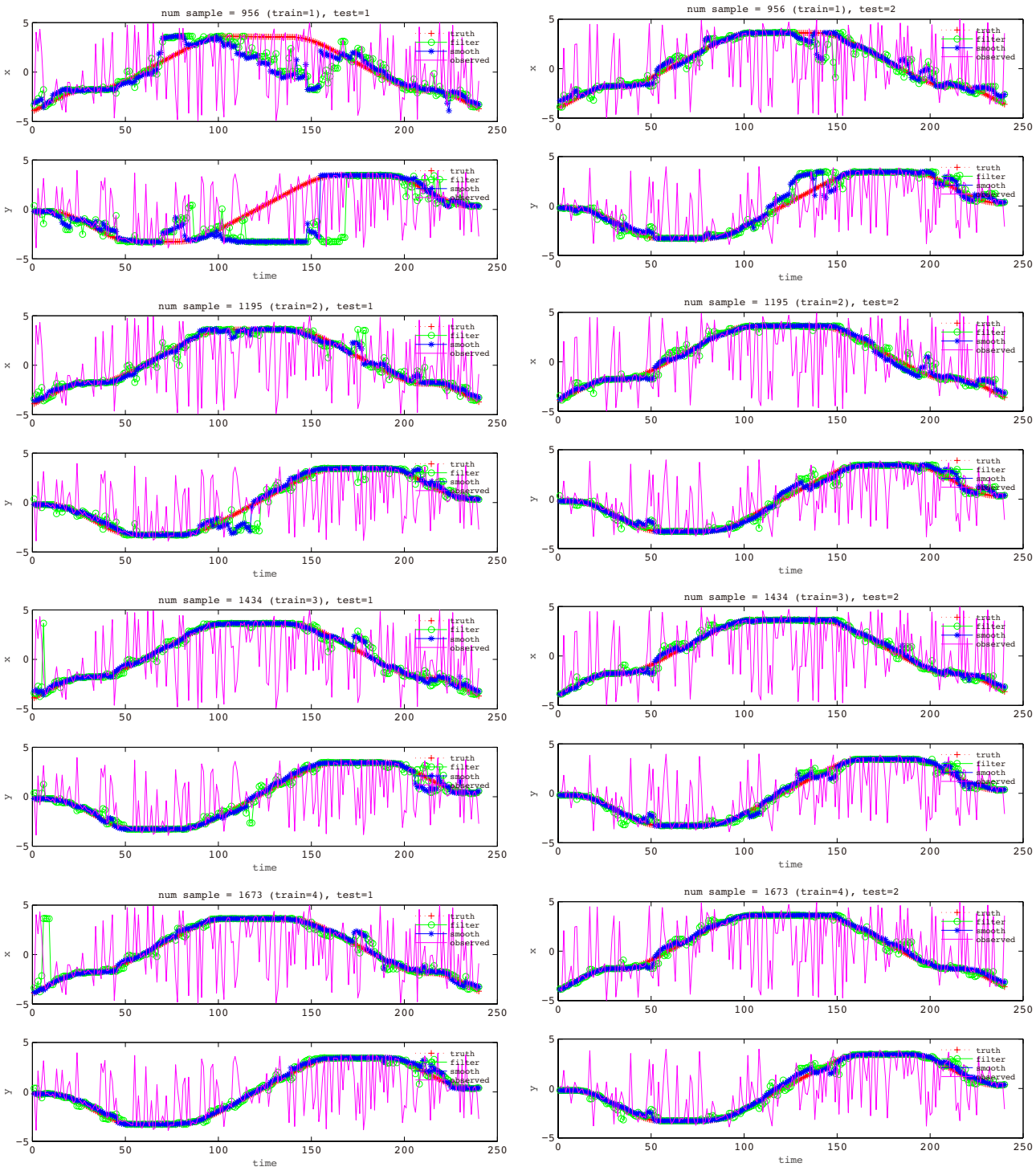

Figure 5: Performance of the nKB-filter and the nKB-smoother in different training and test data on the clutter problem. This figure shows 8 (4×2) experimental results. The upper-left two figures show the performance on the dimension x and y when the training sample size is $n = 956$, respectively. The lower figures show the results when the training sample size is increased to $n = 956, 1195, 1434, 1673$. It is observed that the performance is increased. The right eight figures show results on different test data.

where $1 - \rho$ and ρ are probabilities of measurements from the actual target and clutter, respectively. The measurement from the actual target is a Gaussian $N(\mathbf{z}_t | H\mathbf{x}_t, R)$ with the measurement model matrix H and

noise covariance matrix R. The measurement from the clutter is uniform on the area S. We used the same parameter setting as the RBMCDA's demo used, i.e., the size of time step $\Delta t = 0.1$, $q = 0.1$, $\rho = 1/2$, $S = [-5, 5] \times [-4, 4]$, and

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, R = \begin{pmatrix} 0.05 & 0 \\ 0 & 0.05 \end{pmatrix}.$$

nKB-smoother setting: We used Gaussian kernels $k_{\mathcal{X}}(\mathbf{x}_1, \mathbf{x}_2) = \mathrm{e}^{-||\mathbf{x}_1 - \mathbf{x}_2||^2/2\sigma_{\mathcal{X}}^2}$ and $k_{\mathcal{Z}}(\mathbf{z}_1, \mathbf{z}_2) = \mathrm{e}^{-||\mathbf{z}_1 - \mathbf{z}_2||^2/2\sigma_{\mathcal{Z}}^2}$ for target's states and measurements, respectively, where $\sigma_{\mathcal{X}} = \sigma_{\mathcal{Z}} = 0.1$. We set regularization constants $\epsilon_n = \delta_n = \tilde{\epsilon}_n = \tilde{\delta}_n = 0.001$. Note $\tilde{\epsilon}_n$ and $\tilde{\delta}_n$ are new regularization constants introduced for KB-smoother.

A supplementary video: We present an animation which shows results of the nKB-filter [10] and the nKB-smoother (Algorithm 1) in the clutter problem. Please see a supplementary movie file (.mov). Figure 4 presents a snapshot of the animation at time step $t = 111$.

Supplementary results: Figure 5 shows other results in different training and test data on the clutter problem.

6.3 Marginal Kernel Mean Computation on Tree Graphs

In this section, we present marginal kernel mean computation on general tree graphs by using the nKB-filter and the nKB-smoother, as the extension of state space models.

6.3.1 The nKB-filter & nKB-smoother on N Branch Cases

For ease of understanding, we begin with the two branch case shown in Figure 6 (left). Let $\mathbf{x} := (x_{1:T}, \bar{x}_{t+1:\bar{T}})$ be hidden variables and $\mathbf{z} := (z_{1:T}, \bar{z}_{t+1:\bar{T}})$ be measurement variables. The joint probability density function (pdf) $p(\mathbf{x}, \mathbf{z})$ of Figure 6 (left) is given by[11]

$$p(\mathbf{x}, \mathbf{z}) = \left(\prod_{i=0}^{T-1} p(x_{i+1}|x_i) \right) \left(\prod_{i=1}^{T} p(z_i|x_i) \right) \left(\prod_{i=t}^{\bar{T}-1} p(\bar{x}_{i+1}|\bar{x}_i) \right) \left(\prod_{i=t+1}^{\bar{T}} p(\bar{z}_i|\bar{x}_i) \right),$$

where $p(x_1|x_0) := p(x_1)$ and $\bar{x}_t := x_t$. For ease of presentation, we assume that the transition process $\{p(x_{i+1}|x_i)\}_{i=1}^{T-1}$ and $\{p(\bar{x}_{i+1}|\bar{x}_i)\}_{i=t}^{\bar{T}-1}$ follow the same conditional pdf $p(x'|x)$. We also assume that the measurement process $\{p(z_i|x_i)\}_{i=1}^{T}$ and $\{p(\bar{z}_i|\bar{x}_i)\}_{i=t+1}^{\bar{T}}$ follow the same conditional pdf $p(z|x)$. It is not difficult to extend this to general inhomogenous cases, if there is a training sample for learning each of them. We assume that there are training data $\{\tilde{X}_i, \tilde{X}_i'\}_{i=1}^{l}$ and $\{X_i, Z_i\}_{i=1}^{n}$ for $p(x'|x)$ and $p(z|x)$, respectively.

The objective here is to compute the kernel means $\{m_{X_\tau|\mathbf{z}}\}_{\tau=1}^{T}$ and $\{m_{\bar{X}_\tau|\mathbf{z}}\}_{\tau=t+1}^{\bar{T}}$ of conditional distributions $\{p(x_\tau|\mathbf{z})\}_{\tau=1}^{T}$ and $\{p(\bar{x}_\tau|\mathbf{z})\}_{\tau=t+1}^{\bar{T}}$ given measurements $\mathbf{z}$, respectively. We begin with giving an order to the two branches. Wlog, we set $(x_{t+1:T}, z_{t+1:T}) > (\bar{x}_{t+1:\bar{T}}, \bar{z}_{t+1:\bar{T}})$. We have outputs of the nKB-filter and the nKB-smoother on chain $(x_{1:T}, z_{1:T})$ as

$$\hat{m}_{X_t|z_{1:t}} = \sum_{i=1}^{n} \alpha_i^{(t)} k_{\mathcal{X}}(\cdot, X_i), \quad t = 1, \ldots, T,$$

$$\hat{m}_{X_t|z_{1:T}} = \sum_{i=1}^{l} w_i^{(t)} k_{\mathcal{X}}(\cdot, \tilde{X}_i), \quad t = 1, \ldots, T-1.$$

[11] For simplicity, we omitted illustrations of observable variables $\mathbf{z}$ in Figure 6.

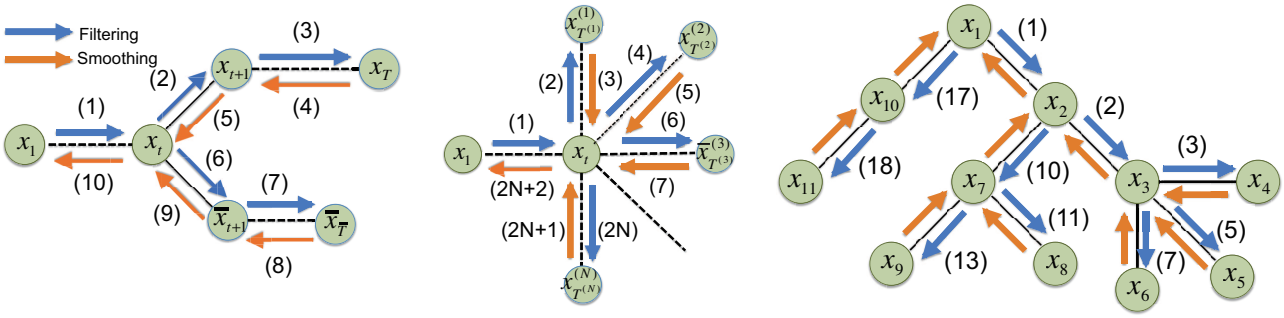

Figure 6: Marginal kernel mean computation on tree graphs using the nKB-filter and the nKB-smoother; (left) the simple two branch case, (middle) general N branch case, and (right) a tree example.

By applying the nonparametric kernel sum rule[12] (Section 2 or Song et al. [25]) to $\hat{m}_{X_t|z_{1:T}}$, we have

$$\hat{m}_{\bar{X}_{t+1}|z_{1:T}} = \hat{\mathcal{U}}_{\bar{X}_{t+1}|X_t}\hat{m}_{X_t|z_{1:T}} = \sum_{i=1}^{l} \eta_i^{(t+1)} k_{\mathcal{X}}(\cdot, \tilde{X}_i'),$$

where $\hat{\mathcal{U}}_{\bar{X}_{t+1}|X_t}$ is the nKSR operator to obtain the estimator $\hat{m}_{\bar{X}_{t+1}|z_{1:T}}$. Next, we apply the KB-filter to the other chain $(\bar{x}_{t+1:\bar{T}}, \bar{z}_{t+1:\bar{T}})$ with the initial belief $\hat{m}_{\bar{X}_{t+1}|z_{1:T}}$, so that the outputs are

$$\hat{m}_{\bar{X}_\tau|z_{1:T},\bar{z}_{t+1:\tau}} = \sum_{i=1}^{n} \bar{\alpha}_i^{(\tau)} k_{\mathcal{X}}(\cdot, X_i), \quad \tau = t+1, \ldots, \bar{T}.$$

Then, we apply the nKB-smoother to the chain $(x_{1:t}, z_{1:t})(\bar{x}_{t+1:\bar{T}}, \bar{z}_{t+1:\bar{T}})$ backward with the initial kernel mean $m_{\bar{X}_{\bar{T}}|\mathbf{z}}$, so that the outputs are

$$\begin{aligned}
\hat{m}_{\bar{X}_\tau|\mathbf{z}} &= \sum_{i=1}^{l} \bar{w}_i^{(\tau)} k_{\mathcal{X}}(\cdot, \tilde{X}_i) \quad \tau = t+1, \ldots, \bar{T}-1. \\
\hat{m}_{X_\tau|\mathbf{z}} &= \sum_{i=1}^{l} \bar{w}_i^{(\tau)} k_{\mathcal{X}}(\cdot, \tilde{X}_i) \quad \tau = 1, \ldots, t.
\end{aligned}$$

The numbers written in Figure 6 (left) show the order of inference of KB-filter and KB-smoother. By induction, the same applies to the N branch case in Figure 6 (middle). First, give an order to the N branches. Then, apply KB-filter and KB-smoother to one branch by one branch. As an example, Figure 6 (right) shows the order of KB-filter and KB-smoother in a tree graph. Thus, the marginal kernel mean computation on a general tree graph is obtained.

[12] By the Markov property, the conditional pdf $p(\bar{x}_{t+1}|z_{1:T})$ has the sum rule expression:

$$\begin{aligned}
p(\bar{x}_{t+1}|z_{1:T}) &= \int p(\tilde{\mathbf{x}}, \tilde{z}_{t+1:\bar{T}}|z_{1:T})\delta(\tilde{x}_{t+1} - \bar{x}_{t+1})d\tilde{z}_{t+1:\bar{T}}d\tilde{\mathbf{x}} \\
&= \int p(\bar{x}_{t+1}|x_t)p(x_t|z_{1:T})dx_t,
\end{aligned}$$

where $\delta(\tilde{x}_{t+1} - \bar{x}_{t+1})$ is the dirac's delta function.