The Nonparametric Kernel Bayes Smoother

Yu Nishiyama Amir Hossein Afsharinejad Shunsuke Naruse
University of Electro-Communications Georgia Tech NTT DATA Mathematical Systems Inc.
Byron Boots Le Song
Georgia Tech 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.

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

\(^1\)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. 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, 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 $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is called a positive-definite kernel if for every $n \in \mathbb{N}$ and $x_1, \ldots, x_n \in \mathcal{X}$, the $n \times n$ 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 $f : \mathcal{X} \rightarrow \mathbb{R}$, which satisfies the following: (i) $k(\cdot, x) \in \mathcal{H}$ for any $x \in \mathcal{X}$, (ii) span $\mathcal{H}_0 := \text{Span}\{k(\cdot, x)\mid x \in \mathcal{X}\}$ is dense in $\mathcal{H}$, i.e., $\mathcal{H} = \bar{\mathcal{H}_0}$, 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 $\mathbb{E}_{X \sim P}[f(X)] := \int_X f(x) dP(x)$ be the expectation of a measurable function $f : \mathcal{X} \rightarrow \mathbb{R}$ 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\footnote{A positive-definite kernel $k$ is bounded if $\sup_{x \in \mathcal{X}} k(x, x) < \infty$.} p.d. kernel $k$ is the RKHS element:

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

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

\footnote{Characteristic kernel is an analogy of the characteristic function $\mathbb{E}_{X \sim P}[e^{i \theta X}]$. Similar to the fact that a characteristic function uniquely specifies a probability distribution.}
$P$ 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:

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

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 $\mathcal{H}_{D_n} := \text{Span}(k(\cdot, x) | x \in D_n)$. We estimate $m_P$ by an estimator $\hat{m}_P = \sum_{i=1}^n w_i k(\cdot, X_i)$ 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 $w := (w_1, \ldots, w_n)^T$ 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 $||\hat{m}_P - m_P||_{\mathcal{H}} \rightarrow 0$ as $n \rightarrow \infty$. 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):

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

**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 $(\mathcal{X}, \mathcal{B}_X)$ and $(\mathcal{Y}, \mathcal{B}_Y)$ be measurable spaces and let $(X, Y)$ be a random variable on $\mathcal{X} \times \mathcal{Y}$ with a joint probability distribution $P_{X,Y}$. Let $P_X$ be the marginal of $P_{X,Y}$ on $\mathcal{X}$ and $P_{Y|X}$ be the conditional distribution given $x \in \mathcal{X}$. We write $P_{Y|X} := \{ P_{Y|X} | x \in \mathcal{X} \}$. Let $P_Y$ be another probability distribution on $\mathcal{X}$ and let $Q_{X,Y}$ be the joint probability distribution given by $P_{Y|X}$ and $P_Y$. Let $Q_Y$ be the marginal of $Q_{X,Y}$ on $\mathcal{Y}$ and let $Q_{X|Y}$ be the conditional distribution of $Q_{X,Y}$ 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 $q(y) = \int_X p(y|x) \pi(x) dx$. 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 $(\mathcal{X}, k_X, \mathcal{H}_X)$ and $(\mathcal{Y}, k_Y, \mathcal{H}_Y)$ be RKHSs. KSR is the computation of the marginal kernel mean $m_{Q_Y} := \mathbb{E}_{Y \sim Q_Y}[k_Y(\cdot, Y)]$ given an input kernel mean $m_H := \mathbb{E}_{X \sim H}[k_X(X, \cdot)]$. KBR is the computation of the posterior kernel mean $m_{Q_{X|Y}} := \mathbb{E}_{X \sim Q_{X|Y}}[k_X(X, \cdot)]$ given $y$ and a prior kernel mean $m_H$. 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 $P_{X,Y}$. Suppose that $H$ is the delta distribution $\delta_\tilde{X}$ on a point $\tilde{X} \in \mathcal{X}$, i.e., $m_H = k_\mathcal{X}(\cdot, \tilde{X})$. Then the conditional kernel mean $m_{P_{X|Y}} = (m_{Q_Y})$ is estimated by $\hat{m}_{P_{X|Y}}(\tilde{X}) = \sum_{i=1}^n w_i k_X(\cdot, X_i)$, where $w = (w_1, \ldots, w_n)^T$ is the kernel ridge regression weights $w = (G_X + n \epsilon_n I_n)^{-1} k_X(\tilde{X})$. Here $G_X = (k_X(X_i, X_j))_{ij} \in \mathbb{R}^{n \times n}$ is the Gram matrix, $\epsilon_n$ is the regularization constant, and $k_X$ is the kernel function of $\mathcal{X}$. For a general input kernel mean estimator $\hat{m}_H := \sum_{i=1}^n \gamma_i k_X(\cdot, X_i)$, output $\hat{m}_{Q_Y}$ is estimated by the weighted sum, i.e., weights $w$ are

$$
w = (G_X + n \epsilon_n I_n)^{-1} G_X \gamma := M^{(nKSR)} \gamma.
$$

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

The nKBR algorithm [10] is obtained by the function-valued kernel ridge regression (nKSR) twice, i.e., regression $Y$ on $X$ with prior kernel mean $m_H$ and regression $X$ on $Y$ with an input observation $y$. Specifically, the nKBR estimates the posterior kernel mean $\hat{m}_{Q_{X|Y}} := \sum_{i=1}^l \tilde{w}_i k_X(\cdot, X_i)$ given an estimator $\hat{m}_{H} := \sum_{i=1}^l \gamma_i k_X(\cdot, \tilde{X}_i)$ and observation $y$ in $\mathcal{Y}$. This is obtained by a matrix multiplication $\tilde{w} = M^{(nKBR)}(\hat{m}_H) k_Y(y)$, where

$$
k_Y(y) := (k_Y(y, Y_1), \ldots, k_Y(y, Y_l))^T \in \mathbb{R}^n
$$

and $M^{(nKBR)}(\hat{m}_H) \in \mathbb{R}^{n \times n}$ is the nKBR matrix depending on $\hat{m}_H$. The nKBR matrix is given by

$$
M^{(nKBR)}(\hat{m}_H) = \text{diag}(w) G_Y ((\text{diag}(w) G_Y)^2 + \delta_n I_n)^{-1} \text{diag}(w)
$$

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

\(^5\)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 \((G_Y)_{ij} = k_Y(Y_i, Y_j)\), and \(\delta_n\) is a regularization constant.

Consistency of estimators \(\hat{m}_{Q_Y}\) and \(\hat{m}_{Q_X|Y}\) when the nKSR and nKBR matrices are used, is proved in Fuku-mizu et al. [10], Song et al. [28].

3 Kernel Bayesian Smoothing

We propose a new addition to existing tools for non-parametric 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 \(X, Z\) 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 \(x \in X\) to the next \(x' \in X\) is a general time-invariant conditional distribution \(P_{X'|X}\). Importantly, we do not make classic assumptions about the distribution \(P_{X'|X}\), e.g. that it has density, is an additive noise model \(x_{t+1} = f(x_t) + \zeta_t\), or that the domain is restricted to Euclidean space \(\mathbb{R}^d\). Instead, we define an RKHS \((X, k_X, \mathcal{H}_X)\) on the domain \(X\). Given transition data \(\{(X_i, X'_i)\}_{i=1}^n\) we nonparametrically learn the conditional kernel mean of \(P_{X'|X}\) from the data.

- **Measurement Process**: The observation of \(z \in Z\) given the hidden state \(x \in X\) is a general time-invariant conditional distribution \(P_{Z|X}\). Again, we do not assume that \(P_{Z|X}\) has a density, is an additive noise model \(z = g(x) + \zeta_s\), or that the domain \(X\) and \(Z\) are restricted to Euclidean spaces. We again define RKHSs \((X, k_X, \mathcal{H}_X)\) and \((Z, k_Z, \mathcal{H}_Z)\) on the domains \(X\) and \(Z\), respectively. We assume data \(\{(X_i, Z_i)\}_{i=1}^n\) and nonparametrically learn the conditional kernel mean of \(P_{Z|X}\) from the data.

In some applications, data \(\{(\tilde{X}_i, \tilde{X}'_i)\}_{i=1}^n\) 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.

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 \(p(x_{l+1}|z_{1:T})\) of past hidden state \(x_{l+1}\) (\(1 \leq t < T\)) given a sequence of observations \(z_{1:T}\). In other words, smoothing gives an accurate estimation of hidden state \(x_{l+1}\) (\(1 \leq t < T\)) by correcting the filtering estimate using the future observations \(z_{l+1:T}\).

Let kernel means of filtering and smoothing distributions be

\[
m_{X_{l}|z_{1:l}} := \mathbb{E}_{X_{l} \sim p(x_l|z_{1:l})}[k_X(\cdot, X_l)], \quad (1 \leq t < T),
\]

\[
m_{X_{l+1}|z_{1:T}} := \mathbb{E}_{X_{l+1} \sim p(x_{l+1}|z_{1:T})}[k_X(\cdot, X_{l+1})], \quad (1 \leq t < T).
\]
We estimate the kernel means in the following non-parametric forms:

\[
m_{X_t|z_1,t} := \sum_{i=1}^{n} \alpha_i(t)k_{X}(\cdot, X_i), \quad (1 \leq t \leq T),
\]

(6)

\[
m_{X_t|z_1,T} := \sum_{i=1}^{l} w_i(t)k_{X}(\cdot, X_i), \quad (1 \leq t < T).
\]

(7)

Weights \(\alpha_i(t) := (\alpha_1(t), \ldots, \alpha_n(t))\) 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:

\[
m_{X_t|z_1,t} := \int k_{X}(x, \cdot) p(x_i|z_1,t)dx_t
\]

\[= \int k_{X}(x, \cdot) \int p(x_i, x_{t+1}|z_1,t)dx_{t+1}dx_t
\]

\[= \int k_{X}(x, \cdot) \int p(x_i|z_{t+1}, z_1,t)p(x_{t+1}|z_1,t)dx_{t+1}dx_t
\]

\[= \int \int k_{X}(x, \cdot)p(x_i|z_{t+1}, z_1,t)p(x_{t+1}|z_1,t)dx_{t+1}dx_t
\]

\[= \int \int m_{X_t|z_{t+1},z_1,t}p(x_{t+1}|z_1,t)dx_{t+1}, (1 \leq t < T),
\]

(8)

where \(m_{X_t|z_{t+1},z_1,t}\) is the conditional kernel mean of \(p(x_{1:t+1}|z_1,t)\). Suppose that \(m_{X_t|z_{t+1},z_1,t}(x) \in H_X\) holds for \(x \in X\) 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:

\[
m_{X_t|z_{1:T}}(x) = \langle m_{X_t|z_1,t}(x), m_{X_{t+1}|z_1,t} \rangle_{H_X},
\]

(9)

where its initialization is the filtering kernel mean \(m_{X_T|z_1:T}\). The conditional kernel mean \(m_{X_t|z_{t+1},z_1,t}\) can be estimated by using the nKB [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

\[
m_{X_t|z_{t+1},z_1,t} = \int k(x, \cdot)p(x_i|z_{t+1}, z_1,t)dx_t
\]

\[= \int k(x, \cdot)\frac{p(x_{t+1}|x_i)p(x_i|z_1,t)}{p(x_{t+1}|z_1,t)}dx_t.
\]

Thus, let \(\tilde{m}_{X_{t+1}|z_1,t}\) be the nKB estimator:

\[
\tilde{m}_{X_t|z_{t+1},z_1,t} = \sum_{i=1}^{l} \gamma_i(t,x_{t+1})k_{X}(\cdot, \tilde{X}_i).
\]

(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 \(\leq t \leq T - 1)\):

- When \(t = T - 1\),

\[
\tilde{m}_{X_{T-1}|z_1,T}(x) = \langle \tilde{m}_{X_{T-1}|z_1}\rangle_{H_X} + \sum_{i=1}^{n} \alpha_i(T)\tilde{m}_{X_{T-1}|z_{1:T-1}}(x)
\]

\[= \sum_{j=1}^{l} \left( \sum_{i=1}^{n} \alpha_i(T)\gamma_j(T-1,X_i) \right)k_{X}(x, \tilde{X}_j)
\]

\[= \sum_{j=1}^{l} w_j(T-1)k_{X}(x, \tilde{X}_j).
\]

(11)

From (11), the weight update is

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

(12)

- Similarly, when \(1 \leq t \leq T - 2\),

\[
\tilde{m}_{X_{t}|z_{1:T}}(x) = \langle \tilde{m}_{X_{t}|z_1,t}(x), \tilde{m}_{X_{t+1}|z_1,t} \rangle_{H_X}
\]

\[= \sum_{j=1}^{l} \left( \sum_{i=1}^{n} w_i(t+1)\gamma_j(t,X_i) \right)k_{X}(x, \tilde{X}_j)
\]

\[= \sum_{j=1}^{l} w_j(t)k_{X}(x, \tilde{X}_j).
\]

(13)

From (13), the weight update is

\[
w_j(t) = \sum_{i=1}^{l} w_i(t+1)\gamma_j(t,X_i), \quad j = 1, \ldots, l.
\]

(14)

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

\[
w(T-1) = \Gamma(T-1)^{\alpha(T)},
\]

\[w(t) = \Gamma(t)w(t+1), \quad (1 \leq t \leq T - 2),
\]

(15)

where

\[
\Gamma(T-1) := (\gamma(T-1,X_1), \ldots, \gamma(T-1,X_n)) \in \mathbb{R}^{l \times n},
\]

\[\Gamma(t) := (\gamma(t,X_1), \ldots, \gamma(t,X_n)) \in \mathbb{R}^{l \times n}.
\]

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 \(x_{t+1} = X_1, \ldots, X_t\)."
The Nonparametric Kernel Bayes Smoother

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

$$\gamma^{(t,x_{i+1})} = M^{(nKBR)}(m_{X|z_{1:i+1}})k_{X'}(x_{t+1}),$$

where $M^{(nKBR)}(m_{X|z_{1:i+1}})$ is the nKBR matrix operating from $X'$ to $X$ given the transition data $(\{X_i, X_i'\})_{i=1}^T$ and filtering prior $m_{X_{t+1}|z_{1:T}}$. We note that the same matrix $M^{(nKBR)}(m_{X_{t+1}|z_{1:t}})$ can be used for computing $\gamma^{(t,x)}$ with different values $x = X_1, \ldots, X_t$. We can use the nKBR matrix (5) for the consistent estimator. Finally, we have the following:

- When $t = T - 1$,
  $\Gamma(T-1) = (\gamma^{(T-1,X_1)}, \ldots, \gamma^{(T-1,X_n)})$
  $= M^{(nKBR)}(m_{X|z_{1:T-1}})G_X G_{X'}$
  $= \text{diag}(\xi_{T-1}G_X)((\text{diag}(\xi_{T-1}G_{X'}))^2 + \delta_n I_n)^{-1}$
  $\times \text{diag}(\xi_{T-1}G_{X'})$

- Similarly, when $1 \leq t \leq T - 2$,
  $\Gamma(t) = (\gamma^{(t,X_1)}, \ldots, \gamma^{(t,X_n)})$
  $= M^{(nKBR)}(m_{X|z_{1:t}})G_X G_{X'}$
  $= \text{diag}(\xi_{t}G_X)((\text{diag}(\xi_{t}G_{X'}))^2 + \delta_n I_n)^{-1}$
  $\times \text{diag}(\xi_{t}G_{X'})$

Here, $\xi_t$ are the nKSR weights:

$\xi_t = (G_X + l\tilde{\xi}I_l)^{-1} m_{X|z_{1:t}} = (G_X + l\tilde{\xi}I_l)^{-1} G_X G_{X'} \alpha^{(t)}$

and $\delta_n$ and $\tilde{\xi}_t$ are their regularization constants.

Algorithm 1 summarizes the nKB-smoother. 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 $\forall f \in \mathcal{H}_X$ over smoothing distribution $p(x_{t+1}|z_{1:T})$ and the mode estimation. The mode is estimated by solving an optimization problem $\hat{x} = \arg \min_{x \in \mathcal{X}} \|\hat{m}_p - k_X(\cdot, x)\|^2 = \arg \max_{x \in \mathcal{X}} \hat{m}_p(x)$ given a smoothing estimator $\hat{m}_p \in \mathcal{H}_X$ [10]. In experiments (Section 4), we report the mode estimation results on the nKBSmoother.

Algorithm 1: The nKB-smoother

**Input:** filtering weights $\alpha^{(1)}, \ldots, \alpha^{(T)}$ and regularization constants $\tilde{\xi}, \delta_n$

**Initialize:** $\Gamma^{(T-1)} = M^{(nKBR)}(m_{X|z_{1:T-1}})G_{X'} X$

$w^{(T-1)} = \Gamma^{(T-1)} \alpha^{(T)}$

for $t = T - 2$ to $t = 1$
do

$\Gamma(t) = M^{(nKBR)}(m_{X|z_{1:t}})G_X G_{X'}$

$w^{(t)} = \Gamma(t) w^{(t+1)}$

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 $O(l^2)$ 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 $O(Tnl^3)$. The cost can be reduced to $O(Tnl^2)$ by 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 $\xi_l(x_{t+1}, \hat{x}_{t+1}) := \langle m_{X|z_{1:t+1}, z_{1:t+1}}, m_{X|z_{1:t+1}, \hat{x}_{t+1}} \rangle_{\mathcal{H}_X}, \Delta m_{X|z_{1:t+1}} := m_{X|z_{1:t+1}, z_{1:t+1}} - \hat{m}_{X|z_{1:t+1}, \hat{x}_{t+1}},$ and $\Delta \xi_l(x_{t+1}, \hat{x}_{t+1}) := \langle \Delta m_{X|z_{1:t+1}}, \Delta m_{X|z_{1:t+1}, \hat{x}_{t+1}} \rangle_{\mathcal{H}_X}$. Assume that $\xi_l \in \mathcal{H}_X \otimes \mathcal{H}_X$. If $\|m_{X|z_{1:t+1}} - \hat{m}_{X|z_{1:t+1}, \hat{x}_{t+1}}\|_{\mathcal{H}_X} = O_p(l^{-\alpha_t})$ for $\alpha_t \in (0, \frac{1}{2})$ and $\|\Delta \xi_l\|_{\mathcal{H}_X} = O_p(l^{-2\beta_t})$ for $\beta_t \in (0, \frac{1}{2})$, then

$\|m_{X|z_{1:t+1}} - \hat{m}_{X|z_{1:t+1}, \hat{x}_{t+1}}\|_{\mathcal{H}_X} = O_p(l^{-\alpha_t})$

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

From the consistency of filtering, it is known [10] that $\|m_{X|z_{1:T}} - \hat{m}_{X|z_{1:T}}\|_{\mathcal{H}_X} = O_p(l^{-\alpha_T})$ for $\alpha_T \in (0, \frac{1}{2})$. By induction with Theorem 3.1, we have the consistency of smoothing $\|m_{X|z_{1:t}} - \hat{m}_{X|z_{1:t}}\|_{\mathcal{H}_X} = O_p(l^{-\alpha_t})$ for any $1 \leq t \leq T$.
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 \(x_t = (x_t, y_t, \dot{x}_t, \dot{y}_t)\) 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 \(z_t = (\hat{x}_t, \hat{y}_t)\) 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 \(\{(x_i, x'_i)\}_{i=1}^n\) and state-observation pairs \(\{(x_i, z_i)\}_{i=1}^n\).

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 \(z_{t+1:T}\) in the kernel mean form.

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 \(\{(x_i, x'_i)\}_{i=1}^n\) and state-observation pairs \(\{(x_i, z_i)\}_{i=1}^n\).

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-

\(^{10}\)Matlab codes and documents are in http://becs.aalto.fi/en/research/bayes/rbmcda/. For details, see the URL and documents within.
The Nonparametric Kernel Bayes Smoother

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 points, 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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MSE on $x_1$</th>
<th>MSE on $x_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalman Filter</td>
<td>0.1082</td>
<td>0.0143</td>
</tr>
<tr>
<td>Kalman Smoothing</td>
<td>0.0121</td>
<td>0.0096</td>
</tr>
<tr>
<td>GP-EKF</td>
<td>1.64128 e-03</td>
<td>1.8133 e-03</td>
</tr>
<tr>
<td>GP-EKS</td>
<td>1.64122 e-03</td>
<td>1.8132 e-03</td>
</tr>
<tr>
<td>nKB-filter</td>
<td>4.3901 e-04</td>
<td>3.1178 e-04</td>
</tr>
<tr>
<td>nKB-smoothing</td>
<td>2.2087e-04</td>
<td>1.9457 e-04</td>
</tr>
</tbody>
</table>

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 semiparametric 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


