Machine Learning (2020) 109:939–972
https://doi.org/10.1007/s10994-019-05852-9

Model-based kernel sum rule: kernel Bayesian inference with probabilistic models

Yu Nishiyama[1] · **Motonobu Kanagawa**[2] · **Arthur Gretton**[3] · **Kenji Fukumizu**[4]

Received: 16 September 2014 / Revised: 7 May 2019 / Accepted: 3 October 2019 / Published online: 2 January 2020

Abstract
Kernel Bayesian inference is a principled approach to nonparametric inference in probabilistic graphical models, where probabilistic relationships between variables are learned from data in a nonparametric manner. Various algorithms of kernel Bayesian inference have been developed by combining kernelized basic probabilistic operations such as the kernel sum rule and kernel Bayes' rule. However, the current framework is fully nonparametric, and it does not allow a user to flexibly combine nonparametric and model-based inferences. This is inefficient when there are good probabilistic models (or simulation models) available for some parts of a graphical model; this is in particular true in scientific fields where "models" are the central topic of study. Our contribution in this paper is to introduce a novel approach, termed the *model-based kernel sum rule* (Mb-KSR), to combine a probabilistic model and kernel Bayesian inference. By combining the Mb-KSR with the existing kernelized probabilistic rules, one can develop various algorithms for hybrid (i.e., nonparametric and model-based) inferences. As an illustrative example, we consider Bayesian filtering in a state space model, where typically there exists an accurate probabilistic model for the state transition process. We propose a novel filtering method that combines model-based inference for the state transition process and data-driven, nonparametric inference for the observation generating process. We empirically validate our approach with synthetic and real-data experiments, the latter being the problem of vision-based mobile robot localization in robotics, which illustrates the effectiveness of the proposed hybrid approach.

Keywords Kernel methods · Probabilistic models · Kernel mean embedding · Kernel Bayesian inference · Reproducing kernel Hilbert spaces · Filtering · State space models

1 Introduction

Kernel mean embedding of distributions (Smola et al. 2007; Song et al. 2013; Muandet et al. 2017) is a framework for representing, comparing and estimating probability distributions using positive definite kernels and the Reproducing Kernel Hilbert Spaces (RKHS). In this framework, all distributions are represented as corresponding elements, called *kernel means*,

Editor: Thomas Gärtner.

Extended author information available on the last page of the article

in an RKHS, and comparison and estimation of distributions are carried out by comparison and estimation of the corresponding kernel means. The Maximum Mean Discrepancy (Gretton et al. 2012) and the Hilbert–Schmidt Independence Criterion (Gretton et al. 2005) are representative examples of approaches based on comparison of kernel means; the former is a distance between probability distributions and the latter is a measure of dependence between random variables, both enjoying empirical successes and being widely employed in the machine learning literature (Muandet et al. 2017, Chapter 3).

Kernel Bayesian inference (Song et al. 2011, 2013; Fukumizu et al. 2013) is a nonparametric approach to Bayesian inference based on estimation of kernel means. In this approach, statistical relationships between any two random variables, say $X \in \mathcal{X}$ and $Z \in \mathcal{Z}$ with $\mathcal{X}$ and $\mathcal{Z}$ being measurable spaces, are nonparametrically learnt from training data consisting of pairs $(X_1, Z_1), \ldots, (X_n, Z_n) \in \mathcal{X} \times \mathcal{Z}$ of instances. The approach is useful when the relationship between X and Z is complicated and thus it is difficult to design an appropriate parametric model for the relationship; it is effective when the modeller instead has good knowledge about similarities between objects in each domain, expressed as similarity functions or kernels of the form $k_{\mathcal{X}}(x, x')$ and $k_{\mathcal{Z}}(z, z')$. For instance, the relationship can be complicated when the structures of the two domains $\mathcal{X}$ and $\mathcal{Z}$ are very different, e.g., $\mathcal{X}$ may be a three dimensional space describing locations, $\mathcal{Z}$ may be a space of images, and the relationship between $X \in \mathcal{X}$ and $Z \in \mathcal{Z}$ is such that Z is a vision image taken at a location X; since such images are highly dependent on the environment, it is not straightforward to provide a model description for that relationship. In this specific example, however, one can define appropriate similarity functions or kernels; the Euclidean distance may provide a good similarity measure for locations, and there are also a number of kernels for images developed in computer vision (e.g., Lazebnik et al. 2006). Given a sufficient number of training examples and appropriate kernels, kernel Bayesian inference enables an algorithm to learn such complicated relationships in a nonparametric manner, often with strong theoretical guarantees (Caponnetto and Vito 2007; Grünewälder et al. 2012a; Fukumizu et al. 2013).

As standard Bayesian inference consists of basic probabilistic rules such as the *sum rule*, *chain rule* and *Bayes' rule*, kernel Bayesian inference consists of kernelized probabilistic rules such as the *kernel sum rule*, *kernel chain rule* and *kernel Bayes' rule* (Song et al. 2013). By combining these kernelized rules, one can develop *fully-nonparametric* methods for various inference problems in probabilistic graphical models, where probabilistic relationships between any two random variables are learnt nonparametrically from training data, as described above. Examples include methods for filtering and smoothing in state space models (Fukumizu et al. 2013; Nishiyama et al. 2016; Kanagawa et al. 2016a), belief propagation in pairwise Markov random fields (Song et al. 2011), likelihood-free inference for simulator-based statistical models (Nakagome et al. 2013; Mitrovic et al. 2016; Kajihara et al. 2018; Hsu and Ramos 2019), and reinforcement learning or control problems (Grünewälder et al. 2012b; Nishiyama et al. 2012; Rawlik et al. 2013; Boots et al. 2013; Morere et al. 2018). We refer to Muandet et al. (2017, Chapter 4) for a survey of further applications. Typical advantages of the approaches based on kernel Bayesian inference are that (i) they are equipped with theoretical convergence guarantees; (ii) they are less prone to suffer from the curse of dimensionality, when compared to traditional nonparametric methods such as those based on kernel density estimation[1] (Silverman 1986); and (iii) they may be applied

[1] Note that kernel density estimation is a classical nonparametric approach studied in the statistics literature, where "kernels" refer to smoothing kernels, but not reproducing kernels in general. One should not confuse this classical approach with kernel mean embeddings, which is rather a new framework for statistical inference developed in the last decade.

to non-standard spaces of structured data such as graphs, strings, images and texts, by using appropriate kernels designed for such structured data (Schölkopf and Smola 2002).

We argue, however, that the fully-nonparametric nature is both an advantage and a *limitation* of the current framework of kernel Bayesian inference. It is an advantage when there is no part of a graphical model for which a good probabilistic model exists, while it becomes a limitation when there does *exist* a good model for some part of the graphical model. Even in the latter case, kernel Bayesian inference requires a user to prepare training data for that part and an algorithm to learn the probabilistic relationship nonparametrically; this is inefficient, given that there already exists a probabilistic model. The contribution of this paper is to propose an approach to making direct use of a probabilistic model in kernel Bayesian inference, when it is available. Before describing this, we first explain below why and when such an approach can be useful.

1.1 Combining probabilistic models and kernel Bayesian inference

An illustrative example is given by the task of *filtering* in *state space models*; see Fig. 1 for a graphical model. A state space model consists of two kinds of variables: *states* $x_1, \ldots, x_t, \ldots, x_T$, which are the unknown quantities of interest, and *observations* $z_1, \ldots, z_t, \ldots, z_T$, which are measurements regarding the states. Here discrete time intervals are considered, and $t = 1, \ldots, T$ denote time indices with T being the number of time steps. The states evolve according to a Markov process determined by the *state transition model* $p(x_{t+1}|x_t)$ describing the conditional probability of the next state x_{t+1} given the current one x_t. The observation z_t at time t is generated depending only on the corresponding state x_t following the *observation model*, the conditional probability of z_t given x_t. The task of filtering is to provide a (probabilistic) estimate of the state x_t at each time t using the observations $z_1, \ldots, z_t$ provided up to that time; this is to be done sequentially for every time step $t = 1, \ldots, T$.

In various scientific fields that study time-evolving phenomena such as climate science, social science, econometrics and epidemiology, one of the main problems is *prediction* (or *forecasting*) of unseen quantities of interest that will realize in the future. Formulated within a state space model, such quantities of interest are defined as states $x_1, x_2, \ldots, x_T$ of the system. Given an estimate of the initial state x_1, predictions of the states $x_2, \ldots, x_T$ in the future are to be made on the basis of the transition model $p(x_{t+1}|x_t)$, often performed in the form of computer simulation. A problem of such predictions is, however, that errors (which may be stochastic and/or numerical) accumulate over time, and predictions of the states increasingly become unreliable. To mitigate this issue, one needs to make corrections to predictions on the basis of available observations $z_1, z_2, \ldots, z_T$ about the states; such procedure is known as *data assimilation* in the literature, and formulated as filtering in the state space model (Evensen 2009).

When solving the filtering problem with kernel Bayesian inference, one needs to express each of the transition model $p(x_{t+1}|x_t)$ and the observation model $p(z_t|x_t)$ by training data: one needs to prepare examples of state-observation pairs $(X_i, Z_i)_{i=1}^n$ for the observation model, and transition examples $(\tilde{X}_i, \tilde{X}'_i)_{i=1}^m$ for the transition model, where $\tilde{X}_i$ denotes a state at a certain time and $\tilde{X}'_i$ the subsequent state (Song et al. 2009; Fukumizu et al. 2013). However, when there already exists a good probabilistic model for state transitions, it is not efficient to re-express the model by examples and learn it nonparametrically. This is indeed the case in the scientific fields mentioned above, where a central topic of study is to provide an accurate but succinct model description for the evolution of the states $x_1, x_2, \ldots, x_T$,

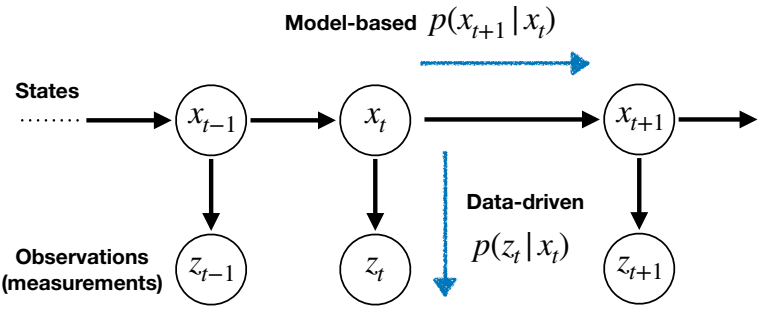

Fig. 1 A graphical description of a state space model, where x_t represent states and z_t observations (or measurements). In this paper we consider a situation where there exists a good probabilistic model for the state-transition probability $p(x_{t+1}|x_t)$, while the observation process $p(z_t|x_t)$ is complicated and to be dealt with in a data-driven, nonparametric way

which may take the form of (ordinary or partial) differential equations or that of multi-agent systems (Winsberg 2010). Therefore it is desirable to make kernel Bayesian inference being able to directly make use of an available transition model in filtering.

1.2 Contributions

Our contribution is to propose a simple yet novel approach to combining the nonparametric methodology of kernel Bayesian inference and model-based inference with probabilistic models. A key ingredient of Bayesian inference in general is the sum rule, i.e., marginalization or integration of variables, which is used for propagating probabilities in graphical models. The proposed approach, termed *Model-based Kernel Sum Rule (Mb-KSR)*, realizes the sum rule in the framework of kernel Bayesian inference, directly making use of an available probabilistic model. (To avoid confusion, we henceforth refer to the kernel sum rule proposed by Song et al. (2009) as the *Nonparametric Kernel Sum Rule (NP-KSR)*.) It is based on analytic representations of conditional kernel mean embeddings (Song et al. 2013), employing a kernel that is compatible with the probabilistic model under consideration. For instance, the use of a Gaussian kernel enables the MB-KSR if the probabilistic model is an additive Gaussian noise model. A richer framework of hybrid (i.e., nonparametric and model-based) kernel Bayesian inference can be obtained by combining the Mb-KSR with existing kernelized probabilistic rules such as the NP-KSR, kernel chain rule and kernel Bayes' rule.

As an illustrative example, we propose a novel method for filtering in a state space model, under the setting discussed in Sect. 1.1 (see Fig. 1). The proposed algorithm is based on hybrid kernel Bayesian inference, realized as a combination of the Mb-KSR and the kernel Bayes' rule. It directly makes use of a transition model $p(x_{t+1}|x_t)$ via the Mb-KSR, while utilizing training data consisting of state-observation pairs $(X_1, Z_1), \ldots, (X_n, Z_n)$ to learn the observation model nonparametrically. Thus it is useful in prediction or forecasting applications where the relationship between observations and states is not easy to be modeled, but examples can be given for it; an example from robotics is given below. This method has an advantage over the fully-nonparametric filtering method based on kernel Bayesian inference (Fukumizu et al. 2013) as it makes use of the transition model $p(x_{t+1}|x_t)$ in a direct manner, without re-expressing it by state transition examples and learning it nonparametrically. This advantage is more significant when the transition model $p(x_{t+1}|x_t)$ is time-dependent (i.e.,

it is not invariant over time); for instance this is the case when the transition model involves control signals, as for the case in robotics.

One illustrative application of our filtering method is mobile robot localization in robotics, which we deal with in Sect. 6. In this problem, there is a robot moving in a certain environment such as a building. The task is to sequentially estimate the positions of the robot as it moves, using measurements obtained from sensors of the robot such as vision images and signal strengths. Thus, formulated as a state space model, the state x_t is the position of the robot, and the observation z_t is the sensor information. The transition model $p(x_{t+1}|x_t)$ describes how the robot's position changes in a short time; since this follows a mechanical law, there is a good probabilistic model such as an odometry motion model (Thrun et al. 2005, Sect. 2.3.2). On the other hand, the observation model $p(z_t|x_t)$ is hard to provide a model description, since the sensor information z_t are highly dependent on the environment and can be noisy; e.g., it may depend on the arrangement of rooms and be affected by people walking in the building. Nevertheless, one can make use of position-sensor examples $(X_1, Z_1), \ldots, (X_n, Z_n)$ collected before the test phase using an expensive radar system or by manual annotation (Pronobis and Caputo 2009).

The remainder of this paper is organized as follows. We briefly discuss related work in Sect. 2 and review the framework of kernel Bayesian inference in Sect. 3. We propose the Mb-KSR in Sect. 4, providing also a theoretical guarantee for it, as manifested in Proposition 1. We then develop the filtering algorithm in Sect. 5. Numerical experiments to validate the effectiveness of the proposed approach are reported in Sect. 6. For simplicity of presentation, we only focus on the Mb-KSR combined with additive Gaussian noise models in this paper, but our framework also allows for other noise models, as described in "Appendix A".

2 Related work

We review here existing methods for filtering in state space methods that are related to our filtering method proposed in Sect. 5. For related work on kernel Bayesian inference, we refer to Sects. 1 and 3.

- The Kalman filters (Kalman 1960; Julier and Uhlmann 2004) and particle methods (Doucet et al. 2001; Doucet and Johansen 2011) are standard approaches to filtering in state space models. These methods typically assume that the domains of states and observations are subsets of Euclidean spaces, and require probabilistic models for both the state transition and observation processes be defined. On the other hand, the proposed filtering method does not assume a probabilistic model for the observation process, and can learn it nonparametrically from training data, even when the domain of observations is a non-Euclidean space.
- Ko and Fox (2009); Deisenroth et al. (2009, 2012) proposed methods for nonparametric filtering and smoothing in state space models based on Gaussian processes (GPs). Their methods nonparametrically learn both the state transition model and the observation model using Gaussian process regression (Rasmussen and Williams 2006), assuming training data are available for the two models. A method based on kernel Bayesian inference has been shown to achieve superior performance compared to GP-based methods, in particular when the Gaussian noise assumption by the GP-approaches is not satisfied (e.g., when noises are multi-modal) (McCalman et al. 2013; McCalman 2013).
- Nonparametric belief propagation (Sudderth et al. 2010), which deals with generic graphical models, nonparametrically estimates the probability density functions of messages

and marginals using kernel density estimation (KDE) (Silverman 1986). In contrast, in kernel Bayesian inference density functions themselves are not estimated, but rather their kernel mean embeddings in an RKHS are learned from data. Song et al. (2011) proposed a belief propagation algorithm based on kernel Bayesian inference, which outperforms nonparametric belief propagation.

- The filtering method proposed by Fukumizu et al. (2013, Sect. 4.3) is fully nonparametric: It nonparametrically learns both the observation process and the state transition process from training data on the basis of kernel Bayesian inference. On the other hand, the proposed filtering method combines model-based inference for the state transition process using an available probabilistic model, and nonparametric kernel Bayesian inference for the observation process.
- The kernel Monte Carlo filter (Kanagawa et al. 2016a) combines nonparametric kernel Bayesian inference with a sampling method. The algorithm generates Monte Carlo samples from a probabilistic model for the state transition process based, and estimates the kernel means of forward probabilities based on them. In contrast, the proposed filtering method does not use sampling but utilizes the analytic expressions of the kernel means of probabilistic models.[2]

3 Preliminaries: nonparametric kernel Bayesian inference

In this section we briefly review the framework of kernel Bayesian inference. We begin by reviewing basic properties of positive definite kernels and reproducing kernel Hilbert spaces (RKHS) in Sect. 3.1, and those of kernel mean embeddings in Sects. 3.2 and 3.3; we refer to Steinwart and Christmann (2008, Sect. 4) for details of the former, and to Muandet et al. (2017, Sect. 3) for those of the latter. We then describe basics of kernel Bayesian inference in Sects. 3.4, 3.5 and 3.6; further details including various applications can be found in Song et al. (2013) and Muandet et al. (2017, Sect. 4).

3.1 Positive definite kernels and reproducing kernel Hilbert space (RKHS)

We first introduce positive definite kernels and RKHSs. Let $\mathcal{X}$ be an arbitrary nonempty set. A symmetric function $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a *positive definite kernel* if it satisfies the following: $\forall n \in \mathbb{N}$ and $\forall x_1, \ldots, x_n \in \mathcal{X}$, the matrix $G \in \mathbb{R}^{n \times n}$ with elements $G_{i,j} = k(x_i, x_j)$ is positive semidefinite. Such a matrix G is referred to as a Gram matrix. For simplicity we may refer to a positive definite kernel k just as a *kernel* in this paper. For instance, kernels on $\mathcal{X} = \mathbb{R}^m$ include the Gaussian kernel $k(x, x') = \exp(-\|x - x'\|^2/\gamma^2)$ and the Laplace kernel $k(x, x') = \exp(-\|x - x'\|/\gamma)$, where $\gamma > 0$.

For each fixed $x \in \mathcal{X}$, $k(\cdot, x)$ denotes a function of the first argument: $x' \to k(x', x)$ for $x' \in \mathcal{X}$. A kernel k is called *bounded* if $\sup_{x \in \mathcal{X}} k(x, x) < \infty$. When $\mathcal{X} = \mathbb{R}^m$, a kernel k called *shift invariant* if there exists a function $\kappa\colon \mathbb{R}^m \to \mathbb{R}$ such that $k(x, x') = \kappa(x - x')$, $\forall x, x' \in \mathbb{R}^m$. For instance, Gaussian, Laplace, Matèrn and inverse (multi-)quadratic kernels are shift-invariant kernels; see Rasmussen and Williams (2006, Sect. 4.2).

[2] Intuitively, the relationship between the kernel Monte Carlo filter and the proposed filter may be understood as something similar to the relationship between a particle filter and a Kalman filter: As the Kalman filter does not require sampling and makes use of the analytic solutions of required integrals, the proposed filter does not perform sampling and uses analytic solutions of the integrals required for computing kernel means.

Let $\mathcal{H}$ be a Hilbert space consisting of functions on $\mathcal{X}$, with $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ being its inner product. The space $\mathcal{H}$ is called a *Reproducing Kernel Hilbert Space (RKHS)*, if there exists a positive definite kernel $k : \mathcal{X} \times \mathcal{X}$ satisfying the following two properties:

$$\begin{aligned} &k(\cdot, x) \in \mathcal{H}, \quad \forall x \in \mathcal{X}, \\ &f(x) = \langle f, k(\cdot, x) \rangle_{\mathcal{H}}, \quad \forall f \in \mathcal{H}, \ \forall x \in \mathcal{X}, \end{aligned} \tag{1}$$

where (1) is called the *reproducing property*; thus k is called the *reproducing kernel* of the RKHS $\mathcal{H}$.

Conversely, for *any* positive definite kernel k, there exists a uniquely associated RKHS $\mathcal{H}$ for which k is the reproducing kernel; this fact is known as the *Moore-Aronszajn theorem* (Aronszajn 1950). Using the kernel k, the associate RKHS $\mathcal{H}$ can be written as the closure of the linear span of functions $k(\cdot, x)$:

$$\mathcal{H} = \overline{\text{span}\,\{k(\cdot, x) : \ x \in \mathcal{X}\}}.$$

3.2 Kernel mean embeddings of distributions

We introduce the concept of kernel mean embeddings of distributions, a framework for representing, comparing and estimating probability distributions using kernels and RKHSs. To this end, let $\mathcal{X}$ be a measurable space and $\mathcal{M}_1(\mathcal{X})$ be the set of all probability distributions on $\mathcal{X}$. Let k be a measurable kernel on $\mathcal{X}$ and $\mathcal{H}$ be the associated RKHS. For any probability distribution $P \in \mathcal{M}_1(\mathcal{X})$, we define its representation in $\mathcal{H}$ as an element called the *kernel mean*, defined as the Bochner integral of $k(\cdot, x) \in \mathcal{H}$ with respect to P:

$$m_P := \int k(\cdot, x) dP(x) \in \mathcal{H}. \tag{2}$$

If k is bounded, then the kernel mean (2) is well-defined and exists for all $P \in \mathcal{M}_1(\mathcal{X})$ (Muandet et al. 2017, Lemma 3.1). Throughout this paper, we thus assume that kernels are bounded. Being an element in $\mathcal{H}$, the kernel mean m_P itself is a function such that $m_P(x') = \int k(x', x) dP(x)$ for $x' \in \mathcal{X}$.

The definition (2) induces a mapping (or embedding; thus the approach is called kernel mean *embedding*) from the set of probability distributions $\mathcal{M}_1(\mathcal{X})$ to the RKHS $\mathcal{H}$: $P \in \mathcal{M}_1(\mathcal{X}) \to m_P \in \mathcal{H}$. If this mapping is one-to-one, that is $m_P = m_Q$ holds if and only if $P = Q$ for $P, Q \in \mathcal{M}_1(\mathcal{X})$, then the reproducing kernel k of $\mathcal{H}$ is called *characteristic* (Fukumizu et al. 2004; Sriperumbudur et al. 2010; Simon-Gabriel and Schölkopf 2018). For example, frequently used kernels on $\mathbb{R}^m$ such as Gaussian, Matérn and Laplace kernels are characteristic; see, e.g., Sriperumbudur et al. (2010); Nishiyama and Fukumizu (2016) for other examples. If k is characteristic, then any $P \in \mathcal{M}_1(\mathcal{X})$ is uniquely associated with its kernel mean m_P; in other words, m_P uniquely identifies the embedded distribution P, and thus m_P contains all information about P. Therefore, when required to estimate certain properties of P from data, one can instead focus on estimation of its kernel mean m_P; this is discussed in Sect. 3.3 below.

An important property regarding the kernel mean (2) is that it is the representer of integrals with respect to P in $\mathcal{H}$: for any $f \in \mathcal{H}$, it holds that

$$\langle m_P, f \rangle_{\mathcal{H}} = \left\langle \int k(\cdot, x) dP(x), f \right\rangle_{\mathcal{H}} = \int \langle k(\cdot, x), f \rangle_{\mathcal{H}} \, dP(x) = \int f(x) dP(x), \tag{3}$$

where the last equality follows from the reproducing property (1). Another important property is that it induces a distance or a metric on the set of probability distributions $\mathcal{M}_1(\mathcal{X})$: A

distance between two distributions $P, Q \in \mathcal{M}_1(\mathcal{X})$ is defined as the RKHS distance between their kernel means $m_P, m_Q \in \mathcal{H}$:

$$\left\| m_P - m_Q \right\|_{\mathcal{H}} = \sup_{\|f\|_{\mathcal{H}} \leq 1} \int f(x) dP(x) - \int f(x) dQ(x),$$

where the expression in the right side is known as the *Maximum Mean Discrepancy (MMD)*; see Gretton et al. (2012, Lemma 4) for a proof of the above identity. MMD is an instance of integral probability metrics, and its relationships to other metrics such as the Wasserstein distance have been studied in the literature (Sriperumbudur et al. 2012; Simon-Gabriel and Schölkopf 2018).

3.3 Empirical estimation of kernel means

In Bayesian inference, one is required to estimate or approximate a certain probability distribution P (or its density function) from data, where P may be a posterior distribution or a predictive distribution of certain quantities of interest. In kernel Bayesian inference, one instead estimates its kernel mean m_P from data; this is justified as long as the kernel k is characteristic.

We explain here how one can estimate a kernel mean in general. Assume that one is interested in estimation of the kernel mean m_P (2). In general, an estimator of m_P takes the form of a weighted sum

$$\hat{m}_P = \sum_{i=1}^{n} w_i k(\cdot, X_i), \tag{4}$$

where $w_1, \ldots, w_n \in \mathbb{R}$ are some weights (some of which can be negative) and $X_1, \ldots, X_n \in \mathcal{X}$ are some points. For instance, assume that one is given i.i.d. sample points $X_1, \ldots, X_n$ from P; then the equal weights $w_1 = \cdots w_n = 1/n$ make (4) is a consistent estimator with convergence rate $\|m_P - \hat{m}_P\|_{\mathcal{H}} = O_p(n^{-\frac{1}{2}})$ (Smola et al. 2007; Tolstikhin et al. 2017). In the setting of Bayesian inference, on the other hand, i.i.d. sample points from the target distribution P are not provided, and thus $X_1, \ldots, X_n$ in (4) cannot be i.i.d. with P. Therefore the weights $w_1, \ldots, w_n$ need to be calculated in an appropriate way depending on the target P and available data; we will see concrete examples in Sects. 3.4, 3.5 and 3.6 below.

From (3), the kernel mean estimate (4) can be used to estimate the integral $\int f(x) dP(x)$ of any $f \in \mathcal{H}$ with respect to P as a weighted sum of function values:

$$\int f(x) dP(x) = \langle m_P, f \rangle_{\mathcal{H}} \approx \langle \hat{m}_P, f \rangle_{\mathcal{H}} = \sum_{i=1}^{n} w_i f(X_i), \tag{5}$$

where the last expression follows from the reproducing property (1). In fact, by the Cauchy-Schwartz inequality, it can be shown that $\left| \int f(x) dP(x) - \sum_{i=1}^{n} w_i f(X_i) \right| \leq \|f\|_{\mathcal{H}} \|\hat{m}_P - m_P\|_{\mathcal{H}}$. Therefore, if $\hat{m}_P$ is a consistent estimator of m_P such that $||\hat{m}_P - m_P||_{\mathcal{H}} \to 0$ as $n \to \infty$, then the weighted sum in (5) is also consistent in the sense that $\left| \int f(x) dP(x) - \sum_{i=1}^{n} w_i f(X_i) \right| \to 0$ as $n \to \infty$. The consistency and convergence rates in the case where f does not belong to $\mathcal{H}$ have also been studied (Kanagawa et al. 2016b, 2019).

3.4 Conditional kernel mean embeddings

For simplicity of presentation, we henceforth assume that probability distributions under consideration have *density functions* with some reference measures; this applies to the rest of this paper. However we emphasize that this assumption is generally not necessary both in practice and theory. This can be seen from how the estimators below are constructed, and from theoretical results in the literature.

We first describe a kernel mean estimator of the form (4) when P is a *conditional* distribution (Song et al. 2009). To describe this, let $\mathcal{X}$ and $\mathcal{Y}$ be two measurable spaces, and let $p(y|x)$ be a conditional density function of $y \in \mathcal{Y}$ given $x \in \mathcal{X}$. Define a kernel $k_{\mathcal{X}}$ on $\mathcal{X}$ and let $\mathcal{H}_{\mathcal{X}}$ be the associated RKHS. Similarly, let $k_{\mathcal{Y}}$ be a kernel on $\mathcal{Y}$ and $\mathcal{H}_{\mathcal{Y}}$ be its RKHS.

Assume that $p(y|x)$ is unknown, but training data $\{(X_i, Y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ approximating it are available; usually they are assumed to be i.i.d. with a joint probability $p(x, y) = p(y|x)p(x)$, where $p(x)$ is some density function on $\mathcal{X}$. Using the training data $\{(X_i, Y_i)\}_{i=1}^n$, we are interested in estimating the kernel mean of the conditional probability $p(y|x)$ on $\mathcal{Y}$ for a given x:

$$m_{\mathcal{Y}|x} := \int k_{\mathcal{Y}}(\cdot, y)p(y|x)dy \in \mathcal{H}_{\mathcal{Y}}, \tag{6}$$

which we call the *conditional kernel mean*.

Song et al. (2009) proposed the following estimator of (6):

$$\begin{aligned} \hat{m}_{\mathcal{Y}|x} &= \sum_{j=1}^n w_j(x)k_{\mathcal{Y}}(\cdot, Y_j), \\ w(x) &:= (w_1(x), \ldots, w_n(x))^\top := (G_X + n\varepsilon I_n)^{-1}\mathbf{k}_{\mathcal{X}}(x) \in \mathbb{R}^n, \end{aligned} \tag{7}$$

where $G_X := (k_{\mathcal{X}}(X_i, X_j))_{i,j=1}^n \in \mathbb{R}^{n\times n}$ is the Gram matrix of $X_1, \ldots, X_n$, $\mathbf{k}_{\mathcal{X}}(x) := (k_{\mathcal{X}}(X_1, x), \ldots k_{\mathcal{X}}(X_n, x))^\top \in \mathbb{R}^n$ quantifies the similarities of x and $X_1, \ldots, X_n$, $I_n \in \mathbb{R}^{n\times n}$ is the identity matrix, and $\varepsilon > 0$ is a regularization constant.

Noticing that the weight vector $w(x)$ in (7) is identical to that of kernel ridge regression or Gaussian process regression (see e.g., Kanagawa et al. 2018, Sect. 3), one can see that (7) is a regression estimator of the mapping from x to the conditional expectation $\int k_{\mathcal{Y}}(\cdot, y)p(y|x)dy$. This insight has been used by Grünewälder et al. (2012a) to show that the estimator (7) is that of *function-valued kernel ridge regression*, and to study convergence rates of (7) by applying results from Caponnetto and Vito (2007). In the context of structured prediction, Weston et al. (2003); Cortes et al. (2005) derived the same estimator under the name of *kernel dependency estimation*, although the connection to embedding of probability distributions was not known at the time.

3.5 Nonparametric kernel sum rule (NP-KSR)

Let $\pi(x)$ be a probability density function on $\mathcal{X}$, and $p(y|x)$ be a conditional density function of $y \in \mathcal{Y}$ given $x \in \mathcal{X}$. Denote by $q(x, y)$ the joint density defined by $\pi(x)$ and $p(y|x)$:

$$q(x, y) := p(y|x)\pi(x), \quad x \in \mathcal{X}, \ y \in \mathcal{Y}. \tag{8}$$

Then the usual *sum rule* is defined as the operation to output the marginal density $q(y)$ on $\mathcal{Y}$ by computing the integral with respect to x:

$$q(y) = \int q(x, y)dx = \int p(y|x)\pi(x)dx. \tag{9}$$

For notational consistency, we write the distribution of $q(y)$ as $Q_{\mathcal{Y}}$.

The Kernel Sum Rule proposed by Song et al. (2009), which we call *Nonparametric Kernel Sum Rule (NP-KSR)* to distinguish it from the Model-based Kernel Sum Rule proposed in this paper, is an estimator of the kernel mean of the marginal density (9):

$$m_{Q_{\mathcal{Y}}} := \int k_{\mathcal{Y}}(\cdot, y)q(y)dy = \int\int k_{\mathcal{Y}}(\cdot, y)p(y|x)\pi(x)dxdy. \tag{10}$$

The NP-KSR estimates this using (i) training data $\{(X_i, Y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ for the conditional density $p(y|x)$ and (ii) a weighted sample approximation $\{(\gamma_i, \tilde{X}_i)\}_{i=1}^\ell \subset \mathbb{R} \times \mathcal{X}$ to the kernel mean $m_\Pi := \int k(\cdot, x)\pi(x)dx$ of the input marginal density $\pi(x)$ of the form

$$\hat{m}_\Pi = \sum_{i=1}^{\ell} \gamma_i k_{\mathcal{X}}(\cdot, \tilde{X}_i), \tag{11}$$

where the subscript Π in the left side denotes the distribution of π. To describe the NP-KSR estimator, it is instructive to rewrite (10) using the conditional kernel means (6) as

$$m_{Q_{\mathcal{Y}}} = \int \left(\int k_{\mathcal{Y}}(\cdot, y)p(y|x)dy \right) \pi(x)dx = \int m_{\mathcal{Y}|x}\pi(x)dx.$$

This implies that this kernel mean can be estimated using the estimator (7) of the conditional kernel means $m_{\mathcal{Y}|x}$ and the weighted sample $\{(\gamma_i, \tilde{X}_i)\}_{i=1}^\ell$, which can be seen as an empirical approximation of the input distribution $\Pi \approx \hat{\Pi} := \sum_{i=1}^\ell \gamma_\ell \delta_{\tilde{X}_i}$, where δ_x denotes the Dirac distribution at $x \in \mathcal{X}$. Thus, the estimator of the NP-KSR is given as

$$\begin{aligned} \mathbf{NP-KSR}: \quad \hat{m}_{Q_{\mathcal{Y}}} &:= \sum_{i=1}^{\ell} \gamma_i \hat{m}_{\mathcal{Y}|\tilde{X}_i} = \sum_{j=1}^{n} w_j k_{\mathcal{Y}}(\cdot, Y_j), \\ w &:= (w_1, \ldots, w_n)^\top := (G_X + n\varepsilon I_n)^{-1} G_{X\tilde{X}}\gamma, \end{aligned} \tag{12}$$

where $\hat{m}_{\mathcal{Y}|\tilde{X}_i}$ is (7) with $x = \tilde{X}_i$, $\gamma := (\gamma_1, \ldots, \gamma_\ell)^\top \in \mathbb{R}^\ell$ and $G_{X\tilde{X}} \in \mathbb{R}^{n\times\ell}$ is such that $(G_{X\tilde{X}})_{i,j} = k_{\mathcal{X}}(X_i, \tilde{X}_j)$. Notice that since $G_{X\tilde{X}}\gamma = (\sum_{j=1}^\ell \gamma_i k_{\mathcal{X}}(X_i, \tilde{X}_j))_{i=1}^n = (\hat{m}_\Pi(X_i))_{i=1}^n$, the weights in (12) can be written as

$$(w_1, \ldots, w_n)^\top = (G_X + n\varepsilon I_n)^{-1}(\hat{m}_\Pi(X_1), \ldots, \hat{m}_\Pi(X_n))^\top. \tag{13}$$

That is, the weights can be calculated in terms of evaluations of the input empirical kernel mean $\hat{m}_\Pi$ at $X_1, \ldots, X_n$; this property will be used in Sect. 4.2.2.

The consistency and convergence rates of the estimator (12), which require the regularization constant ε to decay to 0 as $n \to \infty$ at an appropriate rate, have been studied in the literature (Fukumizu et al. 2013, Theorem 8).

3.6 Kernel Bayes' rule (KBR)

We describe here *Kernel Bayes' Rule (KBR)*, an estimator of of the kernel mean of a posterior distribution (Fukumizu et al. 2013). Let $\pi(x)$ be a prior density on $\mathcal{X}$ and $p(y|x)$ be a

conditional density on $\mathcal{Y}$ given $x \in \mathcal{X}$. The standard Bayes' rule is an operation to produce the posterior density $q(x|y)$ on $\mathcal{X}$ for a given observation $y \in \mathcal{Y}$ induced from $\pi(x)$ and $p(y|x)$:

$$q(x|y) = \frac{\pi(x)p(y|x)}{q(y)}, \quad q(y) := \int \pi(x')p(y|x')dx'.$$

In the setting of KBR, it is assumed that $\pi(x)$ and $p(y|x)$ are unknown but samples approximating them are available; assume that the prior $\pi(x)$ is approximated by weighted points $\{(\gamma_i, \tilde{X}_i)\}_{i=1}^{\ell} \subset \mathbb{R} \times \mathcal{X}$ in the sense that its kernel mean $m_\Pi := \int k_\mathcal{X}(\cdot, x)\pi(x)dx$ is approximated by $\hat{m}_\Pi := \sum_{i=1}^{\ell} \gamma_i k_\mathcal{X}(\cdot, \tilde{X}_i)$ as in (11), and that training data $\{(X_i, Y_i)\}_{i=1}^{n} \subset \mathcal{X} \times \mathcal{Y}$ are provided for the conditional density $p(y|x)$. Using $\hat{m}_\Pi$ and $\{(X_i, Y_i)\}_{i=1}^{n}$, the KBR estimates the kernel mean of the posterior

$$m_{Q_{\mathcal{X}|y}} := \int k_\mathcal{X}(\cdot, x)q(x|y)dx.$$

Specifically the estimator of the KBR is given as follows. Let $w \in \mathbb{R}^n$ be the weight vector defined as (12) or (13), and $D(w) \in \mathbb{R}^{n\times n}$ be a diagonal matrix with its diagonal elements being w. Then the estimator of the KBR is defined by

$$\begin{aligned} \textbf{KBR}: \quad \hat{m}_{Q_{\mathcal{X}|y}} &= \sum_{j=1}^{n} \tilde{w}_j k_\mathcal{X}(\cdot, X_j), \\ \tilde{w} &:= R_{\mathcal{X}|\mathcal{Y}} \mathbf{k}_\mathcal{Y}(y) \in \mathbb{R}^n, \\ R_{\mathcal{X}|\mathcal{Y}} &:= D(w)G_Y\left((D(w)G_Y)^2 + \delta I_n\right)^{-1} \quad D(w) \in \mathbb{R}^{n\times n}, \end{aligned} \tag{14}$$

where $\mathbf{k}_\mathcal{Y}(y) := (k_\mathcal{Y}(y, Y_1), \ldots, k_\mathcal{Y}(y, Y_n))^\top \in \mathbb{R}^n$, $G_Y = (k_\mathcal{Y}(Y_i, Y_j))_{i,j=1}^{n} \in \mathbb{R}^{n\times n}$, and $\delta > 0$ is a regularization constant. This is a consistent estimator: As the number of training data n increases and as $\hat{m}_\Pi$ approaches m_Π, the estimate $\hat{m}_{Q_{\mathcal{X}|y}}$ converges to $m_{Q_{\mathcal{X}|y}}$ under certain assumptions; see Fukumizu et al. (2013, Theorems 6 and 7) for details.

4 Kernel Bayesian inference with probabilistic models

In this section, we introduce the Model-based Kernel Sum Rule (Mb-KSR), a realization of the sum rule in kernel Bayesian inference using a probabilistic model. We describe the Mb-KSR in Sect. 4.1, and show how to combine the MB-KSR and NP-KSR in Sect. 4.2. We explain how the KBR can be implemented when a prior kernel mean estimate is given by a model-based algorithm such as the Mb-KSR in Sect. 4.3. We will use these basic estimators to develop a filtering algorithm for state space models in Sect. 5. As mentioned in Sect. 3.4, we assume that distributions under considerations have density functions for the sake of clarity of presentation.

4.1 Model-based kernel sum rule (Mb-KSR)

Let $\mathcal{X} = \mathcal{Y} = \mathbb{R}^m$ with $m \in \mathbb{N}$. Define kernels $k_\mathcal{X}$ and $k_\mathcal{Y}$ on $\mathcal{X}$ and $\mathcal{Y}$, respectively, and let $\mathcal{H}_\mathcal{X}$ and $\mathcal{H}_\mathcal{Y}$ be their respective RKHSs. Assume that a user defines a probabilistic model as

a conditional density function[3] on $\mathcal{Y}$ given $\mathcal{X}$:

$$p_M(y|x), \quad x, y \in \mathbb{R}^m,$$

where the subscript "M" stands for "Model." Consider the kernel mean of the probabilistic model $p_M(y|x)$:

$$m_{\mathcal{Y}|x} = \int k_{\mathcal{Y}}(\cdot, y) p_M(y|x) dy \in \mathcal{H}_{\mathcal{Y}}, \quad x \in \mathcal{X}. \tag{15}$$

We focus on situations where the above integral has an analytic solution, and thus one can evaluate the value of the kernel mean $m_{\mathcal{Y}|x}(y') = \int k_{\mathcal{Y}}(y', y) p_M(y|x) dy$ for a given $y' \in \mathcal{Y}$.

An example is given by the case where $p_M(y|x)$ is an additive Gaussian noise model, as described in Example 1 below. (Other examples can be found in "Appendix A".) To describe this, let $N(\mu, R)$ be the m-dimensional Gaussian distribution with mean vector $\mu \in \mathbb{R}^m$ and covariance matrix $R \in \mathbb{R}^{m \times m}$, and let $g(x|\mu, R)$ denote its density function:

$$g(x|\mu, R) := |2\pi R|^{-1/2} \exp\left(-(x-\mu)^\top R^{-1}(x-\mu)\right). \tag{16}$$

Then an additive Gaussian noise model is such that an output random variable $Y \in \mathbb{R}^m$ conditioned on an input $x \in \mathcal{X}$ is given as

$$Y = f(x) + \epsilon, \quad \epsilon \sim N(0, \Sigma),$$

where $f : \mathcal{X} \to \mathbb{R}^m$ is a vector-valued function and $\Sigma \in \mathbb{R}^{m \times m}$ is a covariance matrix; or equivalently, the conditional density function is given as

$$p_M(y|x) = g(y|f(x), \Sigma), \quad x, y \in \mathbb{R}^m. \tag{17}$$

The additive Gaussian noise model is ubiquitous in the literature, since the form of the Gaussian density often leads to convenient analytic expressions for quantities of interest. An illustrative example is the Kalman filter (Kalman 1960), which uses linear-Gaussian models for filtering in state space models; in the notation of (17), this corresponds to f being a linear map. Another example is Gaussian process models (Rasmussen and Williams 2006), for which additive Gaussian noises are often assumed with f being a nonlinear function following a Gaussian process.

The following describes how the conditional kernel means can be calculated for additive Gaussian noise models by using Gaussian kernels.

Example 1 *(An additive Gaussian noise model with a Gaussian kernel)* Let $p_M(y|x)$ be an additive Gaussian noise model defined as (17). For a positive definite matrix $R \in \mathbb{R}^{m \times m}$, let $k_R : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ be a normalized Gaussian kernel [4] defined as

$$k_R(x_1, x_2) = g(x_1 - x_2|0, R), \quad x_1, x_2 \in \mathbb{R}^m, \tag{18}$$

[3] For simplicity of presentation we assume the probabilistic model has a density function, but the framework below can also hold even when this assumption does not hold (e.g., when the mapping $x \to y$ is deterministic, in which case the conditional distribution is given with a Dirac delta function).

[4] Here we use a normalized Gaussian kernel $k_R(x_1, x_2) = g(x_1 - x_2|0, R)$ that is of the form of a probability density function, rather than the unnormalized kernel $\tilde{k}_R(x_1, x_2) = \exp(-\frac{1}{2}(x_1 - x_2)^\top R^{-1}(x_1 - x_2))$ standard in the literature (Steinwart and Christmann 2008, p. 153). Our motivation is that, if the normalized kernel is used, then the kernel mean is also of the form of a probability density function, which is convenient since the coefficient is not required to be adjusted. On the other hand, if the unnormalized kernel $\tilde{k}_R$ is used, then the resulting kernel mean should be multiplied by a constant as $\tilde{m}_{\mathcal{Y}|x} = |2\pi R|^{1/2} m_{\mathcal{Y}|x}$, where $|2\pi R|^{1/2}$ is the normalization constant of the Gaussian probability density. We use normalized kernels also for other noise models; see "Appendix A".

where g is the Gaussian density (16). Then the conditional kernel mean (15) with $k_{\mathcal{Y}} := k_R$ is given by

$$m_{\mathcal{Y}|x}(y) = g(y|f(x), \Sigma + R), \quad x, y \in \mathbb{R}^m. \tag{19}$$

Proof For each $x \in \mathcal{X}$, the conditional kernel mean (15) can be written in the form of convolution, $m_{\mathcal{Y}|x}(y) = \int g(y - y'|0, R)g(y'|f(x), \Sigma)dy' =: g(\cdot|0, R) * g(\cdot|f(x), \Sigma)(y)$. and (19) follows from the well-known fact that the convolution of two Gaussian probability densities is given by $g(\cdot|\mu_1, \Sigma_1) * g(\cdot|\mu_2, \Sigma_2) = g(\cdot|\mu_1 + \mu_2, \Sigma_1 + \Sigma_2)$. □

As in Sect. 3.5, let $\pi(x)$ be a probability density function on $\mathcal{X}$ and define the marginal density $q(y)$ on $\mathcal{Y}$ by

$$q(y) = \int p_M(y|x)\pi(x)dx, \quad y \in \mathcal{Y}.$$

The Mb-KSR estimates the kernel mean of this marginal probability

$$m_{Q_{\mathcal{Y}}} := \int k_{\mathcal{Y}}(\cdot, y)q(y)dy = \int \left(\int k_{\mathcal{Y}}(\cdot, y)p_M(y|x)dy \right) \pi(x)dx. \tag{20}$$

This is done by using the probabilistic model $p_M(y|x)$ and an empirical approximation $\hat{m}_\Pi = \sum_{i=1}^{\ell} \gamma_i k_{\mathcal{X}}(\cdot, \tilde{X}_i)$ to the kernel mean $m_\Pi = \int k_{\mathcal{X}}(\cdot, x)\pi(x)dx$ of the input probability $\pi(x)$. Since the weighted points $\{(\gamma_i, \tilde{X}_i)\}_{i=1}^{\ell} \subset \mathbb{R} \times \mathcal{X}$ provide an approximation to the distribution Π of π as $\Pi \approx \hat{\Pi} := \sum_{i=1}^{\ell} \gamma_\ell \delta_{\tilde{X}_i}$, we define the Mb-KSR as follows:

$$\textbf{Mb-KSR}: \quad \hat{m}_{Q_{\mathcal{Y}}} := \sum_{i=1}^{\ell} \gamma_i m_{\mathcal{Y}|\tilde{X}_i} = \sum_{i=1}^{\ell} \gamma_i \int k_{\mathcal{Y}}(\cdot, y)p_M(y|\tilde{X}_i)dy, \tag{21}$$

where $m_{\mathcal{Y}|\tilde{X}_i}$ is the conditional kernel mean (15) with $x := \tilde{X}_i$. In the case of Example 1, for instance, one can compute the value $\hat{m}_{Q_{\mathcal{Y}}}(y)$ for any given $y \in \mathcal{Y}$ by using the analytic expression (19) of $m_{\mathcal{Y}|\tilde{X}_i}$ in (21). As mentioned earlier, however, one can use for the Mb-KSR other noise models by employing appropriate kernels, as described in "Appendix A". One such example is an additive Cauchy noise model with a rational quadratic kernel (Rasmussen and Williams 2006, Eq. 4.19), which should be useful when modeling heavy-tailed random quantities.

We provide convergence results of the Mb-KSR estimator (21), as shown in Proposition 1 below. The proof can be found in "Appendix B". Below O_p is the order notation for convergence in probability, and $\mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}$ denotes the tensor product of two RKHSs $\mathcal{H}_{\mathcal{X}}$ and $\mathcal{H}_{\mathcal{X}}$.

Proposition 1 *Let $\{(\gamma_i, \tilde{X}_i)\}_{i=1}^{\ell} \subset \mathbb{R} \times \mathcal{X}$ be such that $\hat{m}_\Pi := \sum_{i=1}^{\ell} \gamma_i k_{\mathcal{X}}(\cdot, \tilde{X}_i)$, satisfies $||\hat{m}_\Pi - m_\Pi||_{\mathcal{H}_{\mathcal{X}}} = O_p(\ell^{-\alpha})$ as $\ell \to \infty$ for some $\alpha > 0$. For a function $\theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined by $\theta(x, \tilde{x}) := \int \int k_{\mathcal{Y}}(y, \tilde{y})p_M(y|x)p_M(\tilde{y}|\tilde{x})dyd\tilde{y}$ for $(x, \tilde{x}) \in \mathcal{X} \times \mathcal{X}$, assume that $\theta \in \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}$. Then for $m_{Q_{\mathcal{Y}}}$ and $\hat{m}_{Q_{\mathcal{Y}}}$ defined respectively in (20) and (21), we have*

$$\left\| m_{Q_{\mathcal{Y}}} - \hat{m}_{Q_{\mathcal{Y}}} \right\|_{\mathcal{H}_{\mathcal{Y}}} = O_p(\ell^{-\alpha}) \quad (\ell \to \infty).$$

Remark 1 The convergence rate of $\hat{m}_{Q_{\mathcal{Y}}}$ given by the Mb-KSR in Proposition 1 is the same as that of the input kernel mean estimator $\hat{m}_\Pi$. On the other hand, the rate for the NP-KSR is known to become slower than that of the input estimator, because of the need for additional learning and regularization (Fukumizu et al. 2013, Theorem 8). Therefore Proposition 1

shows an advantage of the Mb-KSR over the NP-KSR, when the probabilistic model is correctly specified. The condition that $\theta(\cdot,\cdot) \in \mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}$ is the same as the one made in Fukumizu et al. (2013, Theorem 8).

For any function of the form $f = \sum_{j=1}^m c_j k_\mathcal{Y}(\cdot, y_j) \in \mathcal{H}_\mathcal{Y}$ with $c_1, \ldots, c_m \in \mathbb{R}$ and $y_1, \ldots, y_m \in \mathcal{Y}$, its expectation with respect to $q(y)$ can be approximated using the Mb-KSR estimator (21) as

$$\int f(y)q(y)dy = \sum_{j=1}^m c_j m_{Q_\mathcal{Y}}(y_j) \approx \sum_{j=1}^m c_j \sum_{i=1}^\ell \gamma_i m_{\mathcal{Y}|\tilde{X}_i}(y_j). \tag{22}$$

4.2 Combining the Mb-KSR and NP-KSR

Using the Mb-KSR and NP-KSR, one can perform hybrid (i.e., model-based and nonparametric) kernel Bayesian inference. In the following we describe two examples of such hybrid inference with a simple chain graphical model (Fig. 2). In Sect. 5, we use the estimators derived below corresponding to the two figures in Fig. 2 to develop our filtering algorithm for state space models.

To this end, let $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ be three measurable spaces, and let $k_\mathcal{X}$, $k_\mathcal{Y}$ and $k_\mathcal{Z}$ be kernels defined on the respective spaces. For both of the two cases below, let $\pi(x)$ be a probability density function on $\mathcal{X}$. Assume that we are given weighted points $\{(w_i, \tilde{X}_i)\}_{i=1}^\ell \subset \mathbb{R} \times \mathcal{X}$ that provide an approximation $\hat{m}_\Pi = \sum_{i=1}^\ell \gamma_i k_\mathcal{X}(\cdot, \tilde{X}_i)$ to the kernel mean $\int k_\mathcal{X}(\cdot, x)\pi(x)dx$.

4.2.1 NP-KSR followed by Mb-KSR (Fig. 2, left)

Let $p(y|x)$ be a conditional density function of $y \in \mathcal{Y}$ given $x \in \mathcal{X}$, and $p_M(z|y)$ be a conditional density function of $z \in \mathcal{Z}$ given $y \in \mathcal{Y}$. Suppose that $p(y|x)$ is unknown, but training data $\{(X_i, Y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ for it are available. On the other hand, $p_M(z|y)$ is a probabilistic model, and assume that the kernel $k_\mathcal{Z}$ is chosen so that the conditional kernel mean $m_{\mathcal{Z}|y} := \int k_\mathcal{Z}(\cdot, z)p_M(z|y)$ is analytically computable for each $y \in \mathcal{Y}$. Define marginal densities $q(y)$ on $\mathcal{Y}$ and $q(z)$ on $\mathcal{Z}$ by

$$q(y) := \int \pi(x)p(y|x)dx, \quad q(z) := \int q(y)p_M(z|y)dy,$$

and let $m_{Q_\mathcal{Y}} := \int k_\mathcal{Y}(\cdot, y)q(y)dy$ and $m_{Q_\mathcal{Z}} := \int k_\mathcal{Z}(\cdot, z)q(z)dz$ be their respective kernel means.

The goal here is to estimate $m_{Q_\mathcal{Z}}$ using $\hat{m}_\Pi = \sum_{i=1}^\ell \gamma_i k_\mathcal{X}(\cdot, \tilde{X}_i)$, $\{(X_i, Y_i)\}_{i=1}^n$ and $p_M(z|y)$. This can be done by two steps: (i) first estimate the kernel mean $m_{Q_\mathcal{Y}}$ using the NP-KSR (12) with $\hat{m}_\Pi$ and $\{(X_i, Y_i)\}_{i=1}^n$, obtaining an estimate $\hat{m}_{Q_\mathcal{Y}} = \sum_{j=1}^n w_j k_\mathcal{Y}(\cdot, Y_j)$

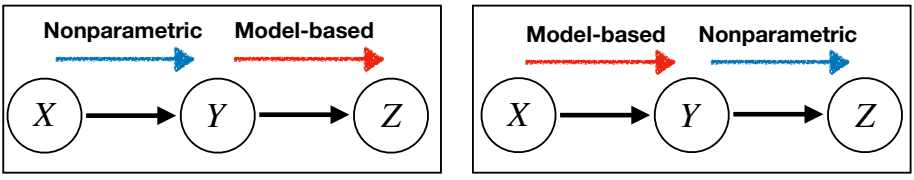

Fig. 2 Hybrid kernel Bayesian inference in a three-variables chain graphical model

with $w := (w_1, \ldots, w_n)^\top := (G_X + n\varepsilon I_n)^{-1} G_{X\tilde{X}}\gamma$, where $\gamma := (\gamma_1, \ldots, \gamma_\ell)^\top \in \mathbb{R}^\ell$, $G_{X\tilde{X}} \in \mathbb{R}^{n\times\ell}$ is such that $(G_{X\tilde{X}})_{i,j} = k_{\mathcal{X}}(X_i, \tilde{X}_j)$ and $\varepsilon > 0$ is a regularization constant; then (ii) apply the Mb-KSR to $\hat{m}_{Q_\mathcal{Y}}$ using $p_M(z|y)$, resulting in the following estimator of $m_{Q_\mathcal{Z}}$:

$$\hat{m}_{Q_\mathcal{Z}} = \sum_{i=1}^{n} w_i m_{\mathcal{Z}|Y_i}, \quad \text{where } m_{\mathcal{Z}|Y_i} := \int k_\mathcal{Z}(\cdot, z) p_M(z|Y_i) dz. \tag{23}$$

4.2.2 Mb-KSR followed by NP-KSR (Fig. 2, right)

Let $p_M(y|x)$ be a conditional density function of $y \in \mathcal{Y}$ given $x \in \mathcal{X}$, and $p(z|y)$ be a conditional density function of $z \in \mathcal{Z}$ given $y \in \mathcal{Y}$. Suppose that for the probabilistic model $p_M(y|x)$, the kernel $k_\mathcal{Y}$ is chosen so that the conditional kernel mean $m_{\mathcal{Y}|x} := \int k_\mathcal{Y}(y|x)dy$ is analytically computable for each $x \in \mathcal{X}$. On the other hand, assume that training data $\{(Y_i, Z_i)\}_{i=1}^n \subset \mathcal{Y} \times \mathcal{Z}$ for the unknown conditional density $p(z|y)$ are available. Define marginal densities $q(y)$ on $\mathcal{Y}$ and $q(z)$ on $\mathcal{Z}$ by

$$q(y) := \int \pi(x) p_M(y|x) dx, \quad q(z) := \int q(y) p(z|y) dy,$$

and let $m_{Q_\mathcal{Y}} := \int k_\mathcal{Y}(\cdot, y) q(y) dy$ and $m_{Q_\mathcal{Z}} := \int k_\mathcal{Z}(\cdot, z) q(z) dz$ be their respective kernel means.

The task is to estimate $m_{Q_\mathcal{Z}}$ using $\hat{m}_\Pi = \sum_{i=1}^\ell \gamma_i k_\mathcal{X}(\cdot, \tilde{X}_i)$, $p_M(y|x)$ and $\{(Y_i, Z_i)\}_{i=1}^n \subset \mathcal{Y} \times \mathcal{Z}$. This can be done by two steps: (i) first estimate the kernel mean $m_{Q_\mathcal{Y}}$ by applying the Mb-KSR (21) to $\hat{m}_\Pi$, yielding an estimate $\hat{m}_{Q_\mathcal{Y}} := \sum_{i=1}^\ell \gamma_i m_{\mathcal{Y}|\tilde{X}_i}$, where $m_{\mathcal{Y}|\tilde{X}_i} = \int k_\mathcal{Y}(\cdot, y) p_M(y|\tilde{X}_i) dy$; (ii) then apply the NP-KSR to $\hat{m}_{Q_\mathcal{Y}}$. To describe (ii), recall that the weights for the NP-KSR can be written as (13) in terms of evaluations of the input empirical kernel mean: thus, the estimator of $m_{Q_\mathcal{Z}}$ by the NP-KSR in (iii) is given by

$$\hat{m}_{Q_\mathcal{Z}} = \sum_{i=1}^{n} w_i k_\mathcal{Z}(\cdot, Z_i), \tag{24}$$

with the weights $w_1, \ldots, w_n$ being

$$\begin{aligned}(w_1, \ldots, w_n)^\top &:= (G_Y + n\varepsilon I_n)^{-1} (\hat{m}_{Q_\mathcal{Y}}(Y_1), \ldots, \hat{m}_{Q_\mathcal{Y}}(Y_n)))^\top \\ &= (G_Y + n\varepsilon I_n)^{-1} G_{Y|\tilde{X}} \gamma,\end{aligned}$$

where $G_{Y|\tilde{X}} \in \mathbb{R}^{n\times\ell}$ is such that $(G_{Y|\tilde{X}})_{ij} = m_{\mathcal{Y}|\tilde{X}_j}(Y_i) = \int k_\mathcal{Y}(Y_i, y) p_M(y|\tilde{X}_i) dy$ and $\gamma := (\gamma_1, \ldots, \gamma_\ell)^\top \in \mathbb{R}^\ell$.

4.3 Kernel Bayes' rule with a model-based prior

We describe how the KBR in Sect. 3.6 can be used when the prior kernel mean $\hat{m}_\Pi$ is given by a model-based estimator such as (21). This way of applying KBR is employed in Sect. 5 to develop our filtering method. The notation in this subsection follows that in Sect. 3.6.

Denote by $\hat{m}_\Pi := \sum_{j=1}^\ell \gamma_j m_j$ a prior kernel mean estimate, where $m_1, \ldots, m_\ell \in \mathcal{H}_\mathcal{X}$ represent model-based kernel mean estimates and $\gamma_1, \ldots, \gamma_\ell \in \mathbb{R}$; for later use, we have written the kernel means $m_1, \ldots, m_\ell$ rather abstractly. For instance, if $\hat{m}_\Pi$ is obtained from

the Mb-KSR (21), then m_j may be given in the form $m_j = \int k_{\mathcal{X}}(\cdot, x) p_M(x|\tilde{X}_j)dx$ for some probabilistic model $p_M(x|\tilde{x})$ and some $\tilde{X}_j \in \mathcal{X}$.

Then the KBR with the prior $\hat{m}_\Pi$ is simply given by the estimator (14) with the weight vector $w \in \mathbb{R}^n$ replaced by the following:

$$w = (G_X + n\varepsilon I_n)^{-1} M\gamma \in \mathbb{R}^n,$$

where $\gamma := (\gamma_1, \ldots, \gamma_n)^\top \in \mathbb{R}^\ell$ and $M \in \mathbb{R}^{n \times \ell}$ is such that $M_{ij} = m_j(X_i)$. This follows from that the weight vector w for the KBR is that of the NP-KSR (13); see also Sect. 4.2.2.

5 Filtering in state space models via hybrid kernel Bayesian inference

Based on the framework for hybrid kernel Bayesian inference introduced in Sect. 4, we propose a novel filtering algorithm for state space models, focusing on the setting of Fig. 1. We formally state the problem setting in Sect. 5.1, and then describe the proposed algorithm in Sect. 5.2, followed by an explanation about how to use the outputs of the proposed algorithm in Sect. 5.3. As before, we assume that all distributions under consideration have density functions for clarity of presentation.

5.1 The problem setting

Let $\mathcal{X}$ be a space of states, and $\mathcal{Z}$ be a space of observations. Let $t = 1, \ldots, T$ denotes the time index with $T \in \mathbb{N}$ being the total number of time steps. A state space model (Fig. 1) consists of two kinds of variables: states $x_1, x_2, \ldots, x_T \in \mathcal{X}$ and observations $z_1, z_2, \ldots, z_T \in \mathcal{Z}$. These variables are assumed to satisfy the conditional independence structure described in Fig. 1, and probabilistic relationships between the variables are specified by two conditional density functions: 1) a transition model $p(x_{t+1}|x_t)$ that describes how the next state x_{t+1} can change from the current state x_t; and 2) an observation model $p(z_t|x_t)$ that describes how likely the observation z_t is generated from the current state x_t. Let $p(x_1)$ be a prior of the initial state x_1.

In this paper, we focus on the case where the transition process is an additive Gaussian noise model, which has been frequently used in the literature. As mentioned before, nevertheless, other noise models described in "Appendix A" can also be used. We consider the following setting.

- **Transition model** Let $\mathcal{X} = \mathbb{R}^m$, and $k_{\mathcal{X}}$ be a Gaussian kernel of the form (18) with covariance matrix $R \in \mathbb{R}^{n \times n}$. Define a vector-valued function $f : \mathbb{R}^m \to \mathbb{R}^m$ and a covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$. It is assumed that f and Σ are provided by a user, and thus known. The transition model is an additive Gaussian noise model such that $x_{t+1} = f(x_t) + \epsilon_t$ with $\epsilon_t \sim N(\mathbf{0}, \Sigma)$, or in the density from,

  $$p(x_{t+1}|x_t) = g(x_{t+1}|f(x_t), \Sigma),$$

  where $g(x|\mu, R)$ denotes the Gaussian density with mean $\mu \in \mathbb{R}^m$ and covariance matrix $R \in \mathbb{R}^{m \times m}$; see (16).
- **Observation model** Let $\mathcal{Z}$ be an arbitrary domain on which a kernel $k_{\mathcal{Z}} : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ is defined. We assume that training data

  $$\{(X_i, Z_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Z}$$

are available for the observation model $p(z_t|x_t)$. The user is not required to have knowledge about the form of $p(z_t|x_t)$.

The task of filtering is to compute the posterior $p(x_t|z_{1:t})$ of the current state x_t given the history of observations $z_{1:t} := (z_1, \ldots, z_t)$ obtained so far; this is to be done sequentially for all time steps $t = 1, \ldots, T$. In our setting, one is required to perform filtering on the basis of the transition model $p(x_{t+1}|x_t)$ and the training data $\{(X_i, Z_i)\}_{i=1}^n$.

Regarding the setting above, note that the training data $\{(X_i, Z_i)\}_{i=1}^n$ are assumed to be available *before* the test phase. This setting appears when *directly measuring the states of the system is possible but requires costs (in terms of computations, time or money) much higher than those for obtaining observations.* For example, in the robot localization problem discussed in Sect. 1.1, it is possible to measure the positions of a robot by using an expensive radar system or by manual annotation, but in the test phase the robot may only be able to use cheap sensors to obtain observations, such as camera images and signal strength information (Pronobis and Caputo 2009). Another example is problems where states can be accurately estimated or recovered from data only available before the test phase. For instance, in tsunami studies (see e.g., Saito 2019), one can recover a tsunami in the past on the basis of data obtained from various sources; however in the test phase, where the task may be that of early warming of a tsunami given that an earthquake has just occurred in an ocean, one can only make use of observations from limited sources, such as seismic intensities and ocean-bottom pressure records.

5.2 The proposed algorithm

In general, a filtering algorithm for a state space model consists of two steps: the *prediction step* and the *filtering step*. We first describe these two steps, as this will be useful in understanding the proposed algorithm.

Assume that the posterior $p(x_{t-1}|z_{1:t-1})$ at time $t-1$ has already been obtained. (If $t = 1$, start from the filtering step below, with $p(x_1|z_{1:0}) := p(x_1)$) In the prediction step, one computes the predictive density $p(x_t|z_{1:t-1})$ by using the sum rule with the transition model $p(x_t|x_{t-1})$:

$$p(x_t|z_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})dx_{t-1}.$$

Suppose then that a new observation z_t has been provided. In the filtering step, one computes the posterior $p(x_t|z_{1:t})$ by using Bayes' rule with $p(x_t|z_{1:t-1})$ as a prior and the observation model $p(z_t|x_t)$ as a likelihood function:

$$p(x_t|z_{1:t}) \propto p(z_t|x_t)p(x_t|z_{1:t-1})$$

Iterations of these two steps over times $t = 1, \ldots, T$ result in a filtering algorithm.

We now describe the proposed algorithm. In our approach, the task of filtering is formulated as estimation of the kernel mean of the posterior $p(x_t|z_{1:t})$:

$$m_{\mathcal{X}_t|z_{1:t}} := \int k_{\mathcal{X}}(\cdot, x_t)p(x_t|z_{1:t})dx_t \in \mathcal{H}_{\mathcal{X}}, \tag{25}$$

which is to be done sequentially for each time $t = 1, \ldots, T$ as a new observation z_t is obtained. (Here $\mathcal{H}_{\mathcal{X}}$ is the RKHS of $k_{\mathcal{X}}$.) The prediction and filtering steps of the proposed algorithm are defined as follows.

Prediction step Let $m_{\mathcal{X}_{t-1}|z_{1:t-1}} \in \mathcal{H}_{\mathcal{X}}$ be the kernel mean of the posterior $p(x_{t-1}|z_{1:t-1})$ at time $t-1$

$$m_{\mathcal{X}_{t-1}|z_{1:t-1}} := \int k_{\mathcal{X}}(\cdot, x_{t-1}) p(x_{t-1}|z_{1:t-1}) dx_{t-1},$$

and assume that its estimate $\hat{m}_{\mathcal{X}_{t-1}|z_{1:t-1}} \in \mathcal{H}_{\mathcal{X}}$ has been computed in the form

$$\hat{m}_{\mathcal{X}_{t-1}|z_{1:t-1}} = \sum_{i=1}^{n} [\boldsymbol{\alpha}_{\mathcal{X}_{t-1}|z_{1:t-1}}]_i k_{\mathcal{X}}(\cdot, X_i), \quad \text{where } \boldsymbol{\alpha}_{\mathcal{X}_{t-1}|z_{1:t-1}} \in \mathbb{R}^n, \tag{26}$$

where $X_1, \ldots, X_n$ are those of the training data. (If $t = 1$, start from the filtering step below.) The task here is to estimate the kernel mean of the predictive density $p(x_t|z_{1:t-1})$:

$$m_{\mathcal{X}_t|z_{1:t-1}} := \int k_{\mathcal{X}}(\cdot, x_t) p(x_t|z_{1:t-1}) dx_t.$$

To this end, we apply the Mb-KSR (Sect. 4.1) to (26) using the transition model $p(x_t|x_{t-1})$ as a probabilistic model: the estimate is given as

$$\hat{m}_{\mathcal{X}_t|z_{1:t-1}} := \sum_{i=1}^{n} [\boldsymbol{\alpha}_{\mathcal{X}_{t-1}|z_{1:t-1}}]_i m_{\mathcal{X}_t|x_t = X_i}, \tag{27}$$

$$m_{\mathcal{X}_t|x_{t-1}=X_i} := \int k_{\mathcal{X}}(\cdot, x_t) p(x_t|x_{t-1} = X_i) dx_t. \tag{28}$$

As shown in Example 1, since both $k_{\mathcal{X}}$ and $p(x_t|x_{t-1})$ are Gaussian, the conditional kernel means (28) have closed form expressions of the form (19).

Filtering step The task here is to estimate the kernel mean (25) of the posterior $p(x_t|z_{1:t})$ by applying the KBR (Sect. 4.3) using (27) as a prior. To describe this, define the kernel mean $m_{\mathcal{Z}_t|z_{1:t-1}} \in \mathcal{H}_{\mathcal{Z}}$ of the predictive density $p(z_t|z_{1:t-1}) := \int p(z_t|x_t) p(x_t|z_{1:t-1}) dx_t$ of a new observation z_t:

$$m_{\mathcal{Z}_t|z_{1:t-1}} := \int k_{\mathcal{Z}}(\cdot, z_t) p(z_t|z_{1:t-1}) dz_t.$$

The KBR first essentially estimates this by applying to the NP-KSR to (27) using the training data $\{(X_i, Z_i)\}_{i=1}^n$; the resulting estimate is

$$\begin{aligned} \hat{m}_{\mathcal{Z}_t|z_{1:t-1}} &:= \sum_{i=1}^{n} [\boldsymbol{\beta}_{\mathcal{Z}_t|z_{1:t-1}}]_i k_{\mathcal{Z}}(\cdot, Z_i), \\ \boldsymbol{\beta}_{\mathcal{Z}_t|z_{1:t-1}} &:= (G_X + n\varepsilon I_n)^{-1} G_{X'|X} \boldsymbol{\alpha}_{\mathcal{X}_{t-1}|z_{1:t-1}} \in \mathbb{R}^n, \end{aligned} \tag{29}$$

where $G_X = (k_{\mathcal{X}}(X_i, X_j))_{i,j=1}^n \in \mathbb{R}^{n \times n}$ and $G_{X'|X} \in \mathbb{R}^{n \times n}$ is defined by evaluations of the conditional kernel means (28): $(G_{X'|X})_{ij} := m_{\mathcal{X}_t|x_{t-1}=X_j}(X_i) = \int k_{\mathcal{X}}(X_i, x_t) p(x_t|x_{t-1} = X_j) dx_t$. (If $t = 1$, generate sample points $\tilde{X}_1, \ldots, \tilde{X}_n \in \mathcal{X}$ i.i.d. from $p(x_1)$, and define $G_{X'|X} \in \mathbb{R}^{n \times n}$ as $(G_{X'|X})_{ij} := k(X_i, \tilde{X}_j)$ and $\boldsymbol{\alpha}_{\mathcal{X}_{t-1}|z_{1:t-1}} := (1/n, \ldots, 1/n)^\top \in \mathbb{R}^n$.)

Using the weight vector $\boldsymbol{\beta}_{Z_t|z_{1:t-1}}$ given above, the KBR then estimates the posterior kernel mean (25) as

$$\hat{m}_{\mathcal{X}_t|z_{1:t}} := \sum_{i=1}^{n} [\boldsymbol{\alpha}_{\mathcal{X}_t|z_{1:t}}]_i k_{\mathcal{X}}(\cdot, X_i), \quad \boldsymbol{\alpha}_{\mathcal{X}_t|z_{1:t}} := R_{\mathcal{X}|\mathcal{Z}}(\boldsymbol{\beta}_{\mathcal{Z}_t|z_{1:t-1}}) \mathbf{k}_Z(z_t), \tag{30}$$

Algorithm 1 The proposed filtering method

Initial Prior Sampling: Generate $\tilde{X}_1, \ldots, \tilde{X}_n \in \mathcal{X}$ i.i.d. from $p(x_1)$, and define $G_{X'|X} \in \mathbb{R}^{n \times n}$ as $(G_{X'|X})_{ij} := k(X_i, \tilde{X}_j)$ and $\boldsymbol{\alpha}_{\mathcal{X}_0|z_{1:0}} := (1/n, \ldots, 1/n)^\top \in \mathbb{R}^n$.
Observe: $z_1 \in \mathcal{Z}$.
Initial Filtering: Compute $\boldsymbol{\alpha}_{\mathcal{X}_1|z_1} \in \mathbb{R}^n$ by the KBR (see Eqs. (29)(30)(31)).
for $t = 2 : T$ **do**
  Compute: $G_{X'|X} \in \mathbb{R}^{n \times n}$ as $(G_{X'|X})_{ij} := \int k_{\mathcal{X}}(X_i, x_t) p(x_t | x_{t-1} = X_j) dx_t$.
  Weight Computation 1: $\boldsymbol{\beta}_{\mathcal{Z}_t|z_{1:t-1}} = (G_X + n\varepsilon I_n)^{-1} G_{X'|X} \boldsymbol{\alpha}_{\mathcal{X}_{t-1}|z_{1:t-1}} \in \mathbb{R}^n$.
  Observe: $z_t \in \mathcal{Z}$.
  Weight Computation 2: $\boldsymbol{\alpha}_{\mathcal{X}_t|z_{1:t}} = R_{\mathcal{X}|\mathcal{Z}}(\boldsymbol{\beta}_{\mathcal{Z}_t|z_{1:t-1}}) \mathbf{k}_Z(z_t) \in \mathbb{R}^n$ (see Eq. (31)).
end for

where $\mathbf{k}_Z(z_t) = (k_{\mathcal{Z}}(Z_i, z_t))_{i=1}^n \in \mathbb{R}^n$ and $R_{\mathcal{X}|\mathcal{Z}}(\boldsymbol{\beta}_{\mathcal{Z}_t|z_{1:t-1}}) \in \mathbb{R}^{n \times n}$ is

$$R_{\mathcal{X}|\mathcal{Z}}(\boldsymbol{\beta}_{\mathcal{Z}_t|z_{1:t-1}}) := D(\boldsymbol{\beta}_{\mathcal{Z}_t|z_{1:t-1}}) G_Z \left((D(\boldsymbol{\beta}_{\mathcal{Z}_t|z_{1:t-1}}) G_Z)^2 + \delta I_n \right)^{-1} D(\boldsymbol{\beta}_{\mathcal{Z}_t|z_{1:t-1}}), \tag{31}$$

where $G_Z := (k_{\mathcal{Z}}(Z_i, Z_j)) \in \mathbb{R}^{n \times n}$ and $D(\boldsymbol{\beta}_{Z_t|z_{1:t-1}}) \in \mathbb{R}^{n \times n}$ is the diagonal matrix with its diagonal elements being $\boldsymbol{\beta}_{\mathcal{Z}_t|z_{1:t-1}}$.

The proposed filtering algorithm is iterative applications of these prediction and filtering steps, as summarized in Algorithm 1. The algorithm results in updating the two weight vectors $\boldsymbol{\beta}_{\mathcal{Z}_t|z_{1:t-1}}, \boldsymbol{\alpha}_{\mathcal{X}_t|z_{1:t}} \in \mathbb{R}^n$.

In Algorithm 1, the computation of the matrix $G_{X'|X}$ is inside the for-loop for $t = 2, \ldots, T$, but one does not need to recompute it if the transition model $p(x_t|x_{t-1})$ is invariant with respect to time t. If the transition model depends on time (e.g., when it involves a control signal), then $G_{X'|X}$ should be recomputed for each time.

5.3 How to use the outputs of Algorithm 1

The proposed filter (Algorithm 1) outputs a sequence of kernel mean estimates $\hat{m}_{\mathcal{X}_1|z_{1:1}}, \hat{m}_{\mathcal{X}_2|z_{1:2}}, \ldots, \hat{m}_{\mathcal{X}_T|z_{1:T}} \in \mathcal{H}_{\mathcal{X}}$ as given in (30), or equivalently a sequence of weight vectors $\boldsymbol{\alpha}_{\mathcal{X}_1|z_{1:1}}, \boldsymbol{\alpha}_{\mathcal{X}_2|z_{1:2}} \ldots, \boldsymbol{\alpha}_{\mathcal{X}_T|z_{1:T}} \in \mathbb{R}^n$. We describe below two ways of using these outputs. Note that these are not the only ways: e.g., one can also generate samples from a kernel mean estimate using the *kernel herding* algorithm (Chen et al. 2010). See Muandet et al. (2017) for other possibilities.

(i) The integral (or the expectation) of a function $f \in \mathcal{H}_{\mathcal{X}}$ with respect to the posterior $p(x_t|z_{1:t})$ can be estimated as (see Sect. 3.3)

$$\begin{aligned} \int f(x_t) p(x_t|z_{1:t}) dx_t &= \langle m_{\mathcal{X}_t|z_{1:t}}, f \rangle_{\mathcal{H}_{\mathcal{X}}} \\ &\approx \langle \hat{m}_{\mathcal{X}_t|z_{1:t}}, f \rangle_{\mathcal{H}_{\mathcal{X}}} = \sum_{i=1}^n [\boldsymbol{\alpha}_{X_t|z_{1:t}}]_i f(X_i). \end{aligned}$$

(ii) A pseudo-MAP (maximum a posteriori) estimate of the posterior $p(x_t|z_{1:t})$ is obtained by solving the preimage problem (see Fukumizu et al. 2013, Sect. 4.1).

$$\hat{x}_t := \arg\min_{x \in \mathcal{X}} \| k_{\mathcal{X}}(\cdot, x) - \hat{m}_{\mathcal{X}_t|z_{1:t}} \|^2_{\mathcal{H}_{\mathcal{X}}}. \tag{32}$$

If for some $C > 0$ we have $k_{\mathcal{X}}(x, x) = C$ for all $x \in \mathcal{X}$ (e.g., when $k_{\mathcal{X}}$ is a shift-invariant kernel), (32) can be rewritten as $\hat{x}_t = \arg\max_{x \in \mathcal{X}} \hat{m}_{\mathcal{X}_t|z_{1:t}}(x)$. If $k_{\mathcal{X}}$ is a Gaussian kernel k_R (as we employ in this paper), then the following recursive algorithm can be used to solve this optimization problem (Mika et al. 1999):

$$x^{(s+1)} = \frac{\sum_{i=1}^{n} X_i [\boldsymbol{\alpha}_{\mathcal{X}_t|z_{1:t}}]_i k_R(X_i, x^{(s)})}{\sum_{i=1}^{n} [\boldsymbol{\alpha}_{\mathcal{X}_t|z_{1:t}}]_i k_R(X_i, x^{(s)})} \quad (s = 0, 1, 2, \ldots). \tag{33}$$

The initial value $x^{(0)}$ can be selected randomly. (Another option may be to set $x^{(0)}$ as a point $X_{i_{\max}} \in \{X_1, \ldots, X_n\}$ in the training data that is associated with the maximum weight (i.e., $i_{\max} = \arg\max[\boldsymbol{\alpha}_{\mathcal{X}_t|z_{1:t}}]_i$).) Note that the algorithm (33) is only guaranteed to converge to the local optimum, if the kernel mean estimate $\hat{m}_{\mathcal{X}_t|z_{1:t}}(x)$ has multiple modes.

6 Experiments

We report three experimental results showing how the use of the Mb-KSR can be beneficial in kernel Bayesian inference when probabilistic models are available. In the first experiment (Sect. 6.1), we deal with simple problems where we can exactly evaluate the errors of kernel mean estimators in terms of the RKHS norm; this enables rigorous empirical comparisons between the Mb-KSR, NP-KSR and combined estimators. We then report results comparing the proposed filtering method (Algorithm 1) to existing approaches by applying them to a synthetic state space model (Sect. 6.2) and to a real data problem of vision-based robot localization in robotics (Sect. 6.3).

6.1 Basic experiments with ground-truths

We first consider the setting described in Sects. 3.5 and 4.1 to compare the Mb-KSR and the NP-KSR. Let $\mathcal{X} = \mathcal{Y} = \mathbb{R}^m$. Define a kernel $k_{\mathcal{X}}$ on $\mathcal{X}$ as a Gaussian kernel $k_{R_{\mathcal{X}}}$ with covariance matrix $R_{\mathcal{X}} \in \mathbb{R}^{n \times n}$ as defined in (18); similarly, let $k_{\mathcal{Y}} = k_{R_{\mathcal{Y}}}$ be a Gaussian kernel on $\mathcal{Y}$ with covariance matrix $R_{\mathcal{Y}} \in \mathbb{R}^{n \times n}$.

Let $p(y|x)$ be a conditional density on $\mathcal{Y}$ given $x \in \mathcal{X}$, which we we define as an additive linear Gaussian noise model: $p(y|x) = g(y|Ax, \Sigma)$ for $x, y \in \mathbb{R}^m$, where $\Sigma \in \mathbb{R}^{m \times m}$ is a covariance matrix and $A \in \mathbb{R}^{m \times m}$. The input density function $\pi(x)$ on $\mathcal{X}$ is defined as a Gaussian mixture $\pi(x) := \sum_{i=1}^{L} \xi_i g(x|\mu_i, W_i)$, where $L \in \mathbb{N}$, $\xi_i \geq 0$ are mixture weights such that $\sum_{i=1}^{L} \xi = 1$, $\mu_i \in \mathbb{R}^m$ are mean vectors and $W_i \in \mathbb{R}^{m \times m}$ are covariance matrices. Then the output density $q(y) := \int p(y|x)\pi(x)dx$ is also a Gaussian mixture $q(y) = \sum_{i=1}^{L} \xi_i g(y|A\mu_i, \Sigma + AW_iA^\top)$.

The task is to estimate the kernel mean $m_{Q_{\mathcal{Y}}} = \int k_{\mathcal{Y}}(\cdot, x)q(y)dy$ of the output density $q(y)$, which has a closed form expression

$$m_{Q_{\mathcal{Y}}} = \sum_{i=1}^{L} \xi_i g(\cdot|A\mu_i, R_{\mathcal{Y}} + \Sigma + AW_iA^\top).$$

This expression is used to evaluate the error $\left\| m_{Q_{\mathcal{Y}}} - \hat{m}_{Q_{\mathcal{Y}}} \right\|_{\mathcal{H}_{\mathcal{Y}}}$ in terms of the distance of the RKHS $\mathcal{H}_{\mathcal{Y}}$, where $\hat{m}_{Q_{\mathcal{Y}}}$ is an estimate given by the Mb-KSR (21) or that of the NP-KSR (12). For the Mb-KSR, the conditional density $p(y|x)$ is treated as a probabilistic model

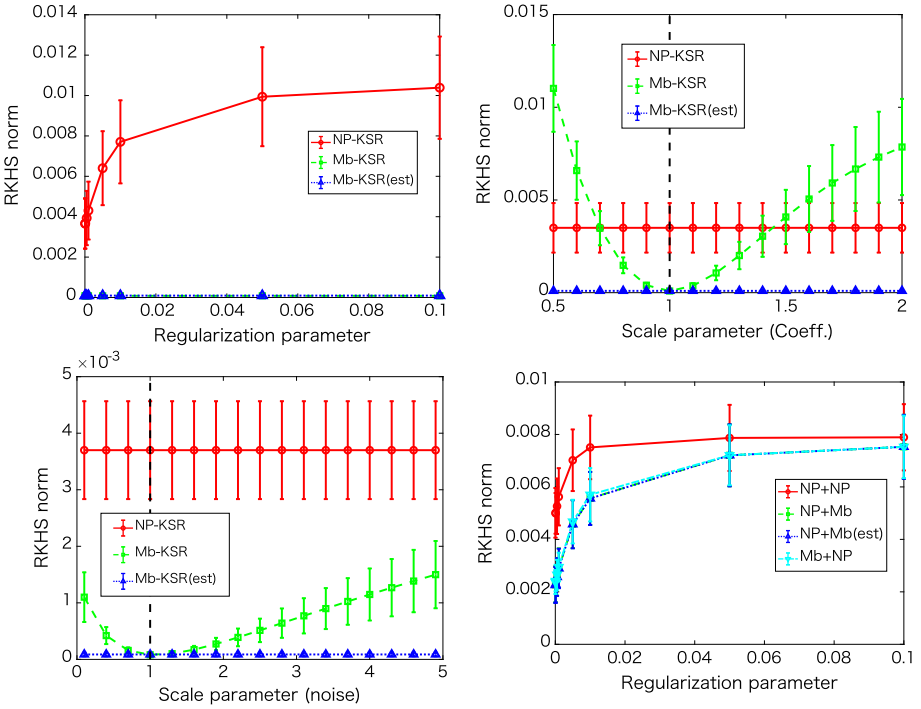

Fig. 3 Top left: estimation errors $\left\|m_{Q_\mathcal{Y}} - \hat{m}_{Q_\mathcal{Y}}\right\|_{\mathcal{H}_\mathcal{Y}}$ versus regularization constants ϵ. The errors of the Mb-KSR and the Mb-KSR (est) are very small and overlap each other. Top right: a model misspecification case (estimation errors vs. scale parameters $\sigma_1 > 0$). Bottom left: a model misspecification case (estimation errors vs. scale parameters $\sigma_2 > 0$). Bottom right: estimation errors $\left\|m_{Q_\mathcal{Z}} - \hat{m}_{Q_\mathcal{Z}}\right\|_{\mathcal{H}_\mathcal{Z}}$ versus regularization constants ϵ for combined estimators. The errors of three estimators (i) NP-KSR and Mb-KSR, (ii) NP-KSR and estimated Mb-KSR and (iii) Mb-KSR and NP-KSR are very close and thus overlap each other. In all the figures, the error bars indicate the standard deviations over 30 independent trials

$p_M(y|x)$, while for the NP-KSR training data are generated for $p(y|x)$; details are explained below.

We performed the following experiment 30 times, independently generating involved data. Fix parameters $m = 2$, $A = \Sigma = I_2$, $L = 4$, $\xi_1 = \cdots = \xi_4 = 1/4$, $R_\mathcal{X} = 0.1 I_2$ and $R_\mathcal{Y} = I_2$. We generated training data $\{(X_i, Y_i)\}_{i=1}^{500}$ for the conditional density $p(y|x)$ by independently sampling from the joint density $p(x, y) := p(y|x)p(x)$, where $p(x)$ is the uniform distribution on $[-10, 10]^2 \subset \mathcal{X}$. The parameters in each component of $\pi(x) = \sum_{i=1}^{L} \xi_i g(x|\mu_i, W_i)$ were randomly generated as $\mu_i \overset{i.i.d.}{\sim} \mathrm{Uni}[-5, 5]^2$ $(i = 1, 2, 3, 4)$ and $W_i = U_i^\top U_i$ with $U_i \overset{i.i.d.}{\sim} \mathrm{Uni}[-2, 2]^4$ $(i = 1, 2, 3, 4)$, where "Uni" denotes the uniform distribution. The input kernel mean $m_\Pi := \int k(\cdot, x)\pi(x)dx$ was then approximated as $\hat{m}_\Pi = \frac{1}{500}\sum_{i=1}^{500} k_\mathcal{X}(\cdot, \tilde{X}_i)$, where $\tilde{X}_1 \ldots \tilde{X}_{500} \in \mathcal{X}$ were generated independently from $\pi(x)$.

Figure 3 (top left) shows the averages and standard deviations of the error $\left\|m_{Q_\mathcal{Y}} - \hat{m}_{Q_\mathcal{Y}}\right\|_{\mathcal{H}_\mathcal{Y}}$ over the 30 independent trials, with the estimate $\hat{m}_{Q_\mathcal{Y}}$ given by three different approaches: NP-KSR, Mb-KSR and "Mb-KSR (est)." The NP-KSR learned $p(y|x)$ with using the training data $\{(X_i, Y_i)\}_{i=1}^{500}$, and we report results with different regularization

constants as $\varepsilon = [.1, .05, .01, .005, .001, .0005, .0001, .00005]$ (horizontal axis). For the Mb-KSR, we used the true $p(y|x)$ as a probabilistic model $p_M(y|x)$. "Mb-KSR (est)" is the Mb-KSR with $p_M(y|x)$ being the linear Gaussian model with parameters A and Σ learnt from $\{(X_i, Y_i)\}_{i=1}^{500}$ by maximum likelihood estimation.

We can make the following observations from Fig. 3 (top left): 1) If the probabilistic model $p_M(y|x)$ is given by parametric learning with a well-specified model, then the performance of the Mb-KSR is as good as that of the Mb-KSR with a correct model; 2) While the NP-KSR is a consistent estimator, its performance is worse than the Mb-KSR, possibly due to the limited sample size and the nonparametric nature of the estimator; 3) The performance of the NP-KSR is sensitive to the choice of a regularization constant.

We next discuss results highlighting the Mb-KSR using misspecified probabilistic models, shown in Fig. 3 (top right and bottom left). Here the NP-KSR used the best regularization constant in Fig. 3 (top left), and the Mb-KSR (est) was given in the same way as above. In Fig. 3 (top right), the Mb-KSR used a misspecified model defined as $p_M(y|x) = g(y|\sigma_1 Ax, \Sigma)$, where $\sigma_1 > 0$ controls the degree of misspecification (horizontal axis); $\sigma_1 = 1$ gives the correct model $p(y|x)$ and is emphasized with the vertical line in the figure. In Fig. 3 (bottom left), the Mb-KSR used a misspecified model $p_M(y|x) = g(y|Ax, \sigma_2\Sigma)$ with $\sigma_2 > 0$; the case $\sigma_2 = 1$ provides the correct model and is indicated by the vertical line. These two figures show the sensitivity of the Mb-KSR to the model specification, but we also observe that the Mb-KSR outperforms the NP-KSR if the degree of misspecification is not severe. The figures also imply that, when it is possible, the parameters in a probabilistic model should be learned from data, as indicated by the performance of the Mb-KSR (est).

Combined estimators Finally, we performed experiments on the combined estimators made of the Mb-KSR and NP-KSR described in Sects. 4.2.1 and 4.2.2; the setting follows that of these sections, and is defined as follows.

Define the third space as $\mathcal{Z} = \mathbb{R}^m$ with $m = 2$, and let $k_\mathcal{Z} := k_{\mathcal{R}_\mathcal{Z}}$ be the Gaussian kernel (18) on $\mathcal{Z}$ with covariance matrix $R_\mathcal{Z} \in \mathbb{R}^{m\times m}$. Let $p(y|x) := g(y|A_1x, \Sigma_1)$ be the conditional density on $\mathcal{Y}$ given $x \in \mathcal{X}$, and $p(z|y) := p(z|A_2y, \Sigma_2)$ be that on $\mathcal{Z}$ given $y \in \mathcal{Y}$, both being additive linear Gaussian noise models, where we set $A_1 = A_2 = \Sigma_1 = \Sigma_2 = I_m \in \mathbb{R}^{m\times m}$. As before, the input density $\pi(x)$ on $\mathcal{X}$ is a Gaussian mixture $\pi(x) = \sum_{i=1}^L \xi_i g(x|\mu_i, W_i)$. Then the output distribution $Q_\mathcal{Z}$ is also a Gaussian mixture with $L = 4$ and $\xi_1 = \cdots = \xi_4 = 1/4$, and parameters $\mu_i \in \mathbb{R}^m$ and $W_i \in \mathbb{R}^{m\times m}$ are randomly generated as $\mu_i \overset{i.i.d.}{\sim} \mathrm{Uni}[-5, 5]^2$ and $W_i = U_i^\top U_i$ with $U_i \overset{i.i.d.}{\sim} \mathrm{Uni}[-2, 2]^4$. Then the output density is given as a Gaussian mixture $q(z) := \int\int p(z|y)p(y|x)\pi(x)dxdy = \sum_{i=1}^L \xi_i g(z|A_2A_1\mu_i, \Sigma_2 + A_2(\Sigma_1 + A_1W_iA_1^\top)A_2^\top)$.

The task is to estimate the kernel mean $m_{Q_\mathcal{Z}} := \int k_\mathcal{Z}(\cdot, x)q(z)dz$, whose closed form expression is given as

$$m_{Q_\mathcal{Z}} = \sum_{i=1}^L \xi_i g\left(\cdot|A_2A_1\mu_i, R_\mathcal{Z} + \Sigma_2 + A_2(\Sigma_1 + A_1W_iA_1^\top)A_2^\top\right).$$

The error $\left\|m_{Q_\mathcal{Z}} - \hat{m}_{Q_\mathcal{Z}}\right\|_{\mathcal{H}_\mathcal{Z}}$ as measured by the norm of the RKHS $\mathcal{H}_\mathcal{Z}$ can then also be computed exactly for a given estimate $\hat{m}_{Q_\mathcal{Z}}$.

Figure 3 (bottom right) shows the averages and standard deviations of the estimation errors over 30 independent trials, computed for four types of combined estimators referred to as "NP + NP," "NP + Mb," "NP+Mb(est)," and "Mb + NP," which are respectively (i) NP-KSR + NP-KSR, (ii) NP-KSR + Mb-KSR, (iii) NP-KSR + Mb-KSR (est), and (iv) Mb-

KSR + NP-KSR. As expected, the model-combined estimators (ii)–(iv) outperformed the full-nonparametric case (i).

6.2 Filtering in a synthetic state space model

We performed experiments on filtering in a synthetic nonlinear state space model, comparing the proposed filtering method (Algorithm 1) in Sect. 5 with the fully-nonparametric filtering method proposed by Fukumizu et al. (2013). The problem setting, described below, is based on that of Fukumizu et al. (2013, Sect. 5.3).

– (**State transition process**) Let $\mathcal{X} = \mathbb{R}^2$ be the state space, and denote by $x_t := (u_t, v_t)^\top \in \mathbb{R}^2$ the state variable at time $t = 1, \ldots, T$. Let $b, M, \eta, \sigma_h > 0$ be constants. Assume that each x_t has an latent variable $\theta_t \in [0, 2\pi]$, which is an angle. The current state x_t then changes to the next state $x_{t+1} := (u_{t+1}, v_{t+1})^\top$ according to the following nonlinear model:

$$(u_{t+1}, v_{t+1})^\top = (1 + b \sin(M\theta_{t+1}))(\cos\theta_{t+1}, \sin\theta_{t+1})^\top + \varsigma_t, \tag{34}$$

where $\varsigma_t \sim N(\mathbf{0}, \sigma_h^2 I_2)$ is an independent Gaussian noise and

$$\theta_{t+1} = \theta_t + \eta \pmod{2ß}. \tag{35}$$

– (**Observation process**) The observation space is $\mathcal{Z} = \mathbb{R}^2$, and let $z_t \in \mathbb{R}^2$ be the observation at time $t = 1, \ldots, T$. Given the current state $x_t := (u_t, v_t)^\top$, the observation z_t is generated as

$$z_t = (\mathrm{sign}(u_t)|u_t|^{\frac{1}{2}}, \mathrm{sign}(v_t)|v_t|^{\frac{1}{2}})^\top + \xi_t,$$

where $\mathrm{sign}(\cdot)$ outputs the sign of its argument, and ξ_t is an independent zero-mean Laplace noise with standard deviation $\sigma_o > 0$.

We used the fully-nonparametric filtering method by Fukumizu et al. (2013, Sect. 4.3) as a baseline, and we refer to it as the *fully-nonparametric kernel Bayesian filter (fKBF)*. As for the proposed filtering method, the fKBR sequentially estimates the posterior kernel means $m_{\mathcal{X}_t|z_{1:t}} = \int k_{\mathcal{X}}(\cdot, x_t) p(x_t|z_{1:t}) dx_t$ $(t = 1, \ldots, T)$ using the KBR in the filtering step. The difference from the proposed filter is that the fKBR uses the NP-KSR (Sect. 3.5) in the prediction step. Thus, a comparison between these two methods reveals how the use of a probabilistic model via the Mb-KSR is beneficial in the context of state space models.

We generated training data $(X_i, Z_i)_{i=1}^n \subset \mathcal{X} \times \mathcal{Z}$ for the observation model as well as those for the transition process $(X_i, X_i')_{i=1}^n \subset \mathcal{X} \times \mathcal{X}$ by simulating the above state space model, where X_i' denotes the state that is one time ahead of X_i. The proposed filter used $(X_i, Z_i)_{i=1}^n$ in the filtering step, and Eqs. (34) and (35) as a probabilistic model in the prediction step. The fKBF used $(X_i, Z_i)_{i=1}^n$ in the filtering step, and $(X_i, X_i')_{i=1}^n$ in the prediction step. For each of these two methods, we defined Gaussian kernels $k_{R_\mathcal{X}}$ and $k_{R_\mathcal{Z}}$ of the form (18) on $\mathcal{X}$ and $\mathcal{Z}$, respectively, where we set $R_\mathcal{X} = \sigma_\mathcal{X}^2 I_2$ and $R_\mathcal{Z} = \sigma_\mathcal{Z}^2 I_2$ for $\sigma_\mathcal{X}, \sigma_\mathcal{Z} > 0$.

For each method, after obtaining an estimate $\hat{m}_{\mathcal{X}_t|z_{1:t}}$ of the posterior kernel mean at each time $t = 1, \ldots, T$, we computed a pseudo-MAP estimate $\hat{x}_t$ using the algorithm (33) in Sect. 5.3, as a point estimate of the true state x_t. We evaluated the performance of each method by computing the mean squared error (MSE) between such point estimates $\hat{x}_t$ and true states x_t. We tuned the hyper parameters in each method (i.e., regularization constants $\delta, \varepsilon > 0$ and kernel parameters $\sigma_\mathcal{X}, \sigma_\mathcal{Y} > 0$) by two-fold cross validation with grid search. We set $T = 100$ for the test phase.

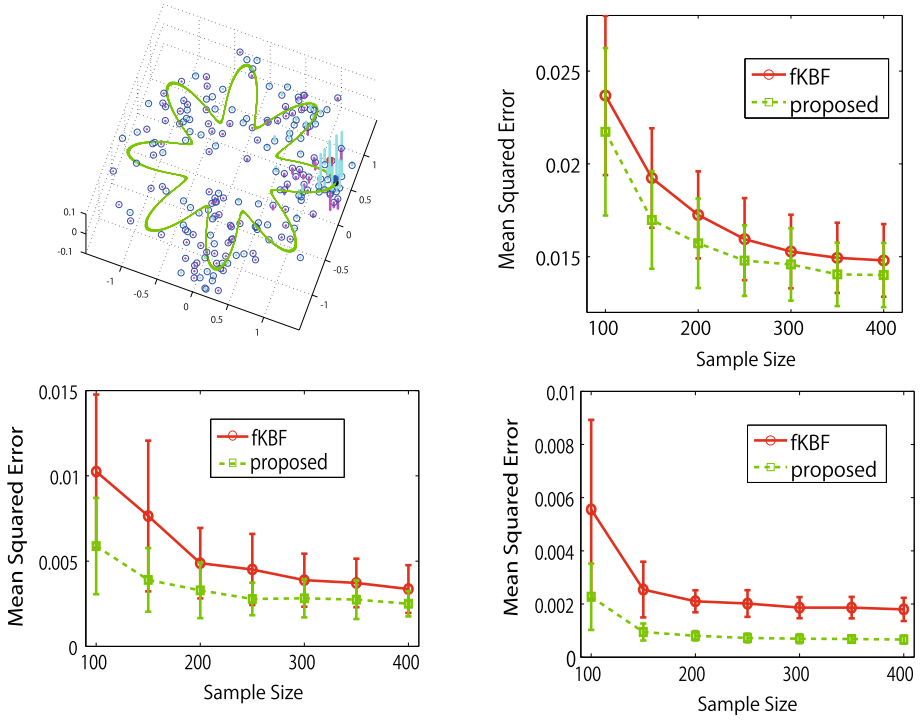

Fig. 4 Comparisons between the proposed filtering method and the fully-nonparametric kernel Bayes filter (fKBF) by Fukumizu et al. (2013). For details, see Sect. 6.2

Figure 4 (top left) visualizes the weight vector $\boldsymbol{\alpha}_{\mathcal{X}_t|z_{1:t}} \in \mathbb{R}^n$ of the estimate $\hat{m}_{X_t|z_{1:t}} = \sum_{i=1}^n [\boldsymbol{\alpha}_{\mathcal{X}_t|z_{1:t}}]_i k_{\mathcal{X}}(\cdot, X_i)$ given by the proposed filter (30) at a certain time point t. In the figure, the green curve is the trajectory of states given by (34) without the noise term. The red and blue points are the observation z_t and the true state x_t. The small points indicate the locations of the training data $X_1, \ldots, X_n$, and the value of the weight $[\boldsymbol{\alpha}_{X_t|z_{1:t}}]_i$ for each data point X_i is plotted in the z axis, where positive and negative weights are colored in cyan and magenta, respectively.

Figure 4 (top right) shows the averages and standard deviations of the MSEs over 30 independent trials for the two methods, where the parameters of the state space model are $b = 0.4$, $M = 8$, $\eta = 1$, $\sigma_h = 0.2$ and $\sigma_o = 0.05$. We performed the experiments for different sample sizes n. As expected, the direct use of the transition process (34) via the Mb-KSR resulted in better performances of the proposed filter than the fully-nonparametric approach.

Similar results are obtained for Fig. 4 (bottom left), where the parameters are set as $b = 0.4$, $M = 8$, $\eta = 1$, $\sigma_o = 0.01$, and the Gaussian noise ς_t in the transition process (34) is replaced by a noise from a Gaussian mixture: $\varsigma_t \sim \frac{1}{4}\sum_{i=1}^4 N(\mu_i, (0.3)^2 I_2)$ with $\mu_1 = (0.2, 0.2)^\top$, $\mu_2 = (0.2, -0.2)^\top$, $\mu_3 = (-0.2, 0.2)^\top$, and $\mu_4 = (-0.2, -0.2)^\top$. We performed this experiment to show the capability of the Mb-KSR to make use of additive mixture noise models (see "Appendix 1").

Finally, Fig. 4 (bottom right) describes results for the case where we changed the transition model in the test phase from that in the training phase. That is, we set $b = 0.4$, $M = 8$,

$\sigma_h = 0.1, \sigma_o = 0.01$ and $\eta = 0.1$ in the training phase, but we changed the parameter η in (35) to $\eta = 0.4$ in the test phase. The proposed filter directly used this knowledge in the test phase by incorporating it by the Mb-KSR, and this resulted in significantly better performances of the proposed filter than the fKBR. Note that such additional knowledge in the test phase is often available in practice, for example in problems where the state transition process involves control signals, as for the case of the robot location problem in the next section. On the other hand, exploitation of such knowledge is not easy for fully nonparametric approaches like fKBR, since they need to express the knowledge in terms of training samples.

6.3 Vision-based robot localization

We performed real data experiments on the vision-based robot localization problem in robotics, formulated as filtering in a state space model. In this problem, we consider a robot moving in a building, and the task is to sequentially estimate the robot's positions in the building in real time, using vision images that the robot has obtained with its camera.

In terms of a state space model, the state x_t at time $t = 1, \ldots, T$ is the robot's position $x_t := (\mathrm{x}_t, \mathrm{y}_t, \theta_t) \in \mathcal{X} := \mathbb{R}^2 \times [-\pi, \pi]$, where $(\mathrm{x}_t, \mathrm{y}_t)$ is the location and θ_t is the direction of the robot, and the observation $z_t \in \mathcal{Z}$ is the vision image taken by the robot at the position x_t. (Here $\mathcal{Z}$ is a space of images.) It is also assumed the robot records odometry data $u_t := (\bar{\mathrm{x}}_t, \bar{\mathrm{y}}_t, \bar{\theta}_t) \in \mathbb{R}^2 \times [-\pi, \pi]$, which are the robot's inner representations of its positions obtained from sensors measuring the revolution of the robot's wheels; such odometry data can be used as control signals (Thrun et al. 2005, Sect. 2.3.2). Thus, the robot localization problem is formulated as the task of filtering using the control signals: estimate the position x_t using a history of vision images $z_1, \ldots, z_t$ and control signals $u_1, \ldots, u_t$ sequentially for every time step $t = 1, \ldots, T$.

The transition model $p(x_{t+1}|x_t, u_t, u_{t+1})$, which includes the odometry data u_t and u_{t+1} as control signals, deals with robot's movements and thus can be modeled on the basis of mechanical laws; we used an odometry motion model [see e.g. Thrun et al. (2005, Sect. 5.4] for this experiment, defined as

$$
\begin{aligned}
\mathrm{x}_{t+1} &= \mathrm{x}_t + \delta_{\mathrm{trans}} \cos(\theta_t + \delta_{\mathrm{rot1}}) + \xi_{\mathrm{x}}, \quad \delta_{\mathrm{rot1}} := \mathrm{atan2}(\bar{\mathrm{y}}_{t+1} - \bar{\mathrm{y}}_t, \bar{\mathrm{x}}_{t+1} - \bar{\mathrm{x}}_t) - \bar{\theta}_t, \\
\mathrm{y}_{t+1} &= \mathrm{y}_t + \delta_{\mathrm{trans}} \sin(\theta_t + \delta_{\mathrm{rot1}}) + \xi_{\mathrm{y}}, \quad \delta_{\mathrm{trans}} := ((\bar{\mathrm{x}}_{t+1} - \bar{\mathrm{x}}_t)^2 + (\bar{\mathrm{y}}_{t+1} - \bar{\mathrm{y}}_t)^2)^{\frac{1}{2}}, \\
\cos\theta_{t+1} &= \cos(\theta_t + \delta_{\mathrm{rot1}} + \delta_{\mathrm{rot2}}) + \xi_c, \qquad \delta_{\mathrm{rot2}} := \bar{\theta}_{t+1} - \bar{\theta}_t - \delta_{\mathrm{rot1}}, \\
\sin\theta_{t+1} &= \sin(\theta_t + \delta_{\mathrm{rot1}} + \delta_{\mathrm{rot2}}) + \xi_s,
\end{aligned}
$$

where $\mathrm{atan2}(\cdot, \cdot)$ is the arctangent function with two arguments, and $\xi_{\mathrm{x}} \sim N(0, \sigma_{\mathrm{x}}^2)$, $\xi_{\mathrm{y}} \sim N(0, \sigma_{\mathrm{y}}^2)$ $\xi_c \sim N(0, \sigma_c^2)$, and $\xi_s \sim N(0, \sigma_s^2)$ are independent Gaussian noises with respective variances $\sigma_{\mathrm{x}}^2, \sigma_{\mathrm{y}}^2, \sigma_c^2$ and σ_s^2, which are the parameters of the transition model.

The observation model $p(z_t|x_t)$ is the conditional probability of a vision image z_t given the robot's position x_t; this is difficult to provide a model description in a parametric form, since it is highly dependent on the environment of the building. Instead, one can use training data $\{(X_i, Z_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Z}$ to provide information of the observation model. Such training data, in general, can be obtained before the test phase, for example by running a robot equipped with expensive sensors or by manually labelling the position X_i for a given image Z_i.

In this experiment we used a publicly available dataset provided by Pronobis and Caputo (2009) designed for the robot localization problem in an indoor office environment. In particular, we used a dataset named *Saarbrücken, Part A, Standard, and Cloudy*. This dataset consists of three similar trajectories that approximately follow the blue dashed path in the

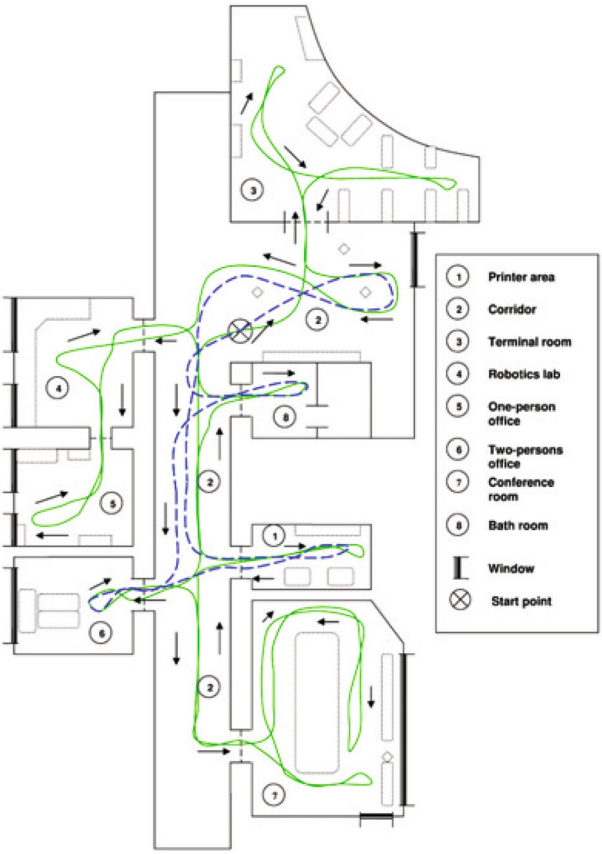

Fig. 5 Paths that a robot approximately followed for data acquisition (Pronobis and Caputo 2009, Fig. 1 (b)) (the use of the figure is granted under the STM Guidelines)

map described in Fig. 5.[5] The three trajectories of the data are plotted in Fig. 6 (left), where each point represents the robot's position $(\mathrm{x}_t, \mathrm{y}_t)$ at a certain time t and the associated arrow the robot's direction θ_t. We used two trajectories for training and the rest for testing.

For our method (and for competing methods that use the transition model), we estimated the parameters σ_{x}^2, σ_{y}^2, σ_c^2 and σ_s^2 in the transition model using the two training trajectories for training by maximum likelihood estimation. As a kernel $k_{\mathcal{Z}}$ on the space $\mathcal{Z}$ of images, we used the spatial pyramid matching kernel (Lazebnik et al. 2006) that is based on the SIFT descriptors (Lowe 2004), where we set the kernel parameters as those recommended by Lazebnik et al. (2006). As a kernel $k_{\mathcal{X}}$ on the space $\mathcal{X}$ of robot's positions, we used a Gaussian kernel. The bandwidth parameters and regularization constants were tuned by two-fold cross validation using the two training trajectories. For point estimation of the position x_t at each time $t = 1, \ldots, T$ in the test phase, we used the position $X_{i_{\max}}$ in the training data $\{(X_i, Z_i)\}$ associated with the maximum in the weights $\boldsymbol{\alpha}_{\mathcal{X}_t|z_{1:t}}$ for the posterior kernel mean estimate (30): $i_{\max} = \arg\max_{i=1,\ldots,n} [\boldsymbol{\alpha}_{\mathcal{X}_t|z_{1:t}}]_i$

[5] Copyright @ 2009, SAGE Publications.

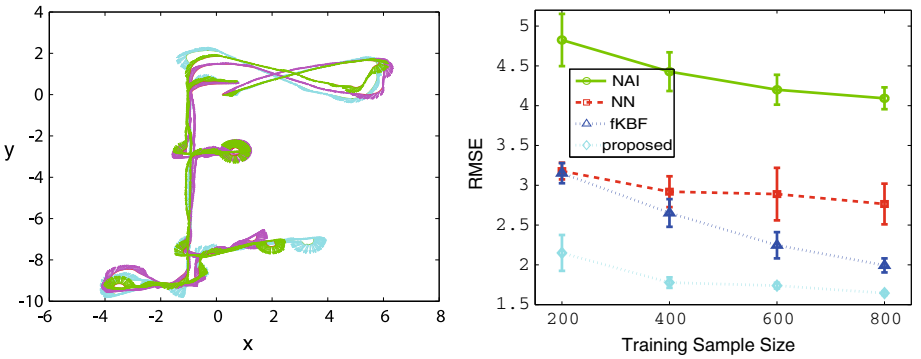

Fig. 6 (Left) Data for three similar trajectories corresponding to the blue path shown in Fig. 5. (x, y) indicates the position of the robot, and the arrow at each position indicates the angle, θ, of the robot's pose. (Right) Estimation accuracy of the robot's position as a function of training sample size n (Color figure online)

We compared the proposed filter with the following three approaches, for which we also tuned hyper-parameters by cross-validation:

- **Naïve method (NAI)** This is a simple algorithm that estimates the robot's position x_t at each time $t = 1, \ldots, T$ as the position $X_{i_{\max}}$ in the training data that is associated with the image $Z_{i_{\max}}$ closest to the given observation z_t in terms of the spatial pyramid matching kernel: $i_{\max} := \arg\max_{i=1,\ldots,n} k_{\mathcal{Z}}(z_t, Z_i)$. This algorithm does not take into account the time-series structure of the problem, was used as a baseline.
- **Nearest neighbors (NN)** (Vlassis et al. 2002): This method uses the k-NN (nearest neighbors) approach to nonparametrically learn the observation model from training data $\{(X_i, Z_i)\}_{i=1}^n$. For the k-NN search we also used the spatial pyramid matching kernel. Filtering is realized by applying a particle filter, using the learned observation model and the transition model (the odometry motion model). Since the learning of the observation model involves a certain heuristic, this approach may produce biases.
- **Fully-nonparametric kernel Bayes filter (fKBF)** (Fukumizu et al. 2013): For an explanation of this method, see Sect. 6.2. Since the NP-KSR, which learns the transition model, involves the control signals (i.d., odometry data), we also defined a Gaussian kernel on controls. As in Sect. 6.2, a comparison between this method and the proposed filter reveals the effect of combining the model-based and nonparametric approaches.

Figure 6 (right) describes averages and standard deviations of RMSEs (root mean squared errors) between estimated and true positions over 10 trials, performed for different training data sizes n. The NN outperforms the NAI, as the NAI does not use the time-series structure of the problem. The fKBF shows performances superior to the NN in particular for larger training data sizes, possibly due to the fact that the fKBF is a statistically consistent approach. The proposed method outperforms the fKBR in particular for smaller training data sizes, showing that the use of the odometry motion model is effective. The result supports our claim that if a good probabilistic model is available, then one should incorporate it into kernel Bayesian inference.

7 Conclusions and future directions

We proposed a method named the model-based kernel sum rule (Mb-KSR) for computing forward probabilities using a probabilistic model in the framework of kernel mean embeddings. By combining it with other basic rules such as the nonparametric kernel sum rule and the kernel Bayes rule (KBR), one can develop inference algorithms that incorporate available probabilistic models into nonparametric kernel Bayesian inference. We specifically proposed in this paper a novel filtering algorithm for a state space model by combining the Mb-KSR and KBR, focusing on the setting where the transition model is available while the observation model is unknown and only state-observation examples are available. We empirically investigated the effectiveness of the proposed approach by numerical experiments that include the vision-based mobile robot localization problem in robotics.

One promising future direction is to investigate applications of the proposed filtering method (or more generally the proposed hybrid approach) in problems where the evolution of states is described by (partial or ordinary) differential equations. This is a situation common in physical scientific fields where the primal aim is to provide model descriptions for time-evolving phenomena, such as climate science, social science, econometrics and epidemiology. In such a problem, a discrete-time state space model is obtained by discretization of continuous differential equations, and the transition model $p(x_{t+1}|x_t)$, which is probabilistic, characterizes numerical uncertainties caused by discretization errors. Importantly, certain numerical solvers of differential equations based on *probabilistic numerical methods* (Hennig et al. 2015; Cockayne et al. 2019; Oates and Sullivan 2019) provide the transition model $p(x_{t+1}|x_t)$ in terms of Gaussian probabilities (Schober et al. 2014; Kersting and Hennig 2016; Schober et al. 2018; Tronarp et al. 2018). Hence, we expect that it is possible to use a transition model obtained from such probabilistic solvers with the Mb-KSR, and to combine a time-series model described by differential equations with nonparametric kernel Bayesian inference.

Another future direction is to extend the proposed filtering method to the *smoothing* problem, where the task is to compute the posterior probability over state trajectories, $p(x_1, \ldots, x_T|z_1, \ldots, z_T)$. This should be possible by incorporating the the Mb-KSR into the fully-nonparametric filtering method based on kernel Bayesian inference developed by Nishiyama et al. (2016). An important issue related to the smoothing problem is that of estimating the parameters of a probabilistic model in hybrid kernel Bayesian inference. For instance, in the smoothing problem, one may also be asked to estimate the parameters in the transition model from a given test sequence of observations. We expect that this can be done by developing an EM-like algorithm, or by using the ABC-based approach to maximum likelihood estimation proposed by Kajihara et al. (2018).

Acknowledgements We would like to thank the anonymous reviewers for their comments that helped us improve the clarity and the quality of the paper. A part of this work was conducted when YN and MK belonged to the Institute of Statistical Mathematics, Tokyo. This research was partly supported by JSPS KAKENHI (B) 22300098, MEXT Grant-in-Aid for Scientific Research on Innovative Areas 25120012, JSPS Wakate (B) 26870821 and the ERC action StG 757275/PANAMA.

Appendix: A conditional kernel means for additive noise models

While we focused on the additive Gaussian model with a Gaussian kernel in the main body, we collect here other noise models and the corresponding kernels that can be used with the Mb-KSR. The key is how to find pairs of a probability density p and a kernel k, both of which are defined on $\mathbb{R}^m$, such that the the kernel mean $m_p(x) := \int k(x, y)p(y)dy$ has a closed form expression. To this end, we briefly mention Nishiyama and Fukumizu (2016), who study certain such pairs.

The idea of Nishiyama and Fukumizu (2016) is to find pairs of a density p and a shift-invariant kernel k such that both p and k share the same functional form; such pairs are called *conjugate*. Recall that a kernel k is shift-invariant if there exists a function $\kappa : \mathbb{R}^m \to \mathbb{R}$ such that $k(x, y) = \kappa(x - y)$ for $x, y \in \mathbb{R}^m$; see Rasmussen and Williams (2006, Section 4.2) for examples of such kernels. In this case the kernel mean m_p can be written as the convolution between κ and p: $m_p(x) = (\kappa * p)(x) = \int \kappa(x - y)p(y)dy$. Therefore one can find pairs of k and p that admit a closed form expression of m_p by examining a convolution semigroup (i.e., a family of density functions that is closed under convolution) in which the function κ is included. For instance, the set of Gaussian densities is closed under convolution, and therefore the kernel mean $m_p = \kappa * p$ of a Gaussian density p has a closed form expression (which is again Gaussian) if κ is also Gaussian.

Examples other than those described below may be found in Table 1 of Briol et al. (2019), which collects pairs of a kernel and a density whose kernel means have closed form expressions.

A.1 Cauchy noise models and rational quadratic kernels

Let $\mu \in \mathbb{R}^m$ and $\Sigma \in \mathbb{R}^{m\times m}$ be a positive definite matrix. The density function of a Cauchy distribution on $\mathbb{R}^m$ (with μ and Σ being its location and scale parameters) is defined as

$$p_{\mathrm{Cauchy}}(x|\mu, \Sigma) = C_{m,\Sigma}(1 + (x - \mu)^\top \Sigma^{-1}(x - \mu))^{-\frac{1+m}{2}}, \tag{36}$$

where $C_{m,\Sigma} := \frac{\Gamma((1+m)/2)}{\Gamma(1/2)\pi^{m/2}|\Sigma|^{1/2}}$ is the normalization constant. Let $f : \mathbb{R}^m \to \mathbb{R}^m$ be a known function. Then an additive Cauchy noise model, which is a conditional density function on $\mathcal{Y} = \mathbb{R}^m$ given $x \in \mathcal{X} = \mathbb{R}^m$, is defined as

$$p_M(y|x) := p_{\mathrm{Cauchy}}(y|f(x), \Sigma). \tag{37}$$

For a positive definite matrix $R \in \mathbb{R}^{m\times m}$, denote by $k_R : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ be a normalized rational quadratic kernel (Rasmussen and Williams 2006, Eq. 4.19) defined as

$$k_R(x_1, x_2) = p_{\mathrm{Cauchy}}(x_1 - x_2|0, R), \quad x_1, x_2 \in \mathbb{R}^m,$$

where p_{Cauchy} is the Cauchy density (36). This kernel can be written as a scale mixture of Gaussian kernels with different bandwidth parameters; see Rasmussen and Williams (2006, p. 87). Then, if $R = \gamma^2\Sigma$, the conditional kernel mean (15) with $k_\mathcal{Y} := k_R$ is given by

$$m_{\mathcal{Y}|x}(y) = p_{\mathrm{Cauchy}}(y|f(x), (1 + \gamma)^2\Sigma), \quad x, y \in \mathbb{R}^m.$$

See Nishiyama and Fukumizu (2016, Example 4.3) for details and for a generalization to α-stable distributions.

A.2 Variance-gamma noise models and Matérn kernels

For $\lambda > m/2$, $\alpha > 0$, $\mu \in \mathbb{R}^m$ and a positive definite matrix $\Sigma \in \mathbb{R}^{m\times m}$, define a *variance-gamma distribution* on $\mathbb{R}^m$ as

$$p_{\mathrm{VG}}(x|\lambda,\alpha,\mu,\Sigma) := \frac{2^{1-\lambda}}{(2\pi)^{m/2}\Gamma(\lambda)}\alpha^{\lambda+m/2}\left[(x-\mu)^\top\Sigma^{-1}(x-\mu)\right]^{(\lambda-m/2)/2} \times K_{\lambda-m/2}\left(\alpha\left[(x-\mu)^\top\Sigma^{-1}(x-\mu)\right]^{1/2}\right), \quad x\in\mathbb{R}^m,$$

where $K_{\lambda-m/2}$ is the modified Bessel function of third kind with index $\lambda-m/2$; this is obtained as a specific case of Hammerstein (2010, Eq. 2.4, p.74) with the asymmetry parameter $\beta = 0$. Note that for $\lambda = (m+1)/2$ and $\alpha = 1$, the variance gamma distribution reduces to a Laplace distribution.

The form of the variance-gamma distributions is the same as that of Matérn kernels (Matèrn 1986). In fact, the Matérn kernel described in Rasmussen and Williams (2006, Eq. 4.14) is, up to constant, given by

$$k(x_1,x_2) = p_{\mathrm{VG}}(x_1-x_2|\nu+m/2,\sqrt{2\nu},0,\Sigma), \quad x_1,x_2\in\mathbb{R}^m, \tag{38}$$

where $\nu > 0$; $\nu + m/2$ is the order of differentiability of functions in the associated RKHS (which is norm-equivalent to a Sobolev space). Note also that the Laplace kernel is the Matérn kernel with $\nu = 1/2$.

For a known function $f : \mathbb{R}^m \to \mathbb{R}^m$, we define an additive variance-gamma noise model as

$$p_M(y|x) := p_{\mathrm{VG}}(y|\lambda,\sqrt{2\nu},f(x),\Sigma).$$

Then with the Matérn kernel (38), the conditional kernel mean (15) is given by

$$m_{\mathcal{Y}|x}(y) = p_{\mathrm{VG}}(y|\lambda+\nu+m/2,\sqrt{2\nu},f(x),\Sigma), \quad x,y\in\mathbb{R}^m.$$

See Nishiyama and Fukumizu (2016, Example 4.6) for details.

A.3 Mixture noise models

For a known function $f : \mathbb{R}^m \to \mathbb{R}^m$, consider a probabilistic model

$$p_M(y|x) = p_{\mathrm{mix}}(y-f(x)), \quad x,y\in\mathbb{R}^m, \tag{39}$$

where p_{mix} is a mixture density

$$p_{\mathrm{mix}}(y) = \sum_{i=1}^{L}\omega_i p_i(y), \quad y\in\mathbb{R}^m,$$

with $\omega_1,\ldots,\omega_L \geq 0$ are mixing coefficients such that $\sum_{i=1}^{L}\omega_i = 1$ and $p_1,\ldots,p_L$ are probability density functions on $\mathbb{R}^m$. For a kernel $k_{\mathcal{Y}}$ on $\mathbb{R}^m$, the conditional kernel mean of the mixture model (39) is then given by

$$m_{\mathcal{Y}|x}(y) = \int k_{\mathcal{Y}}(\cdot,y)p_{\mathrm{mix}}(y-f(x))dy = \sum_{i=1}^{L}\omega_i\int k_{\mathcal{Y}}(\cdot,y)p_i(y-f(x))dy.$$

Therefore, if the terms $\int k_{\mathcal{Y}}(\cdot, y) p_i(y - f(x))dy$ admit closed form expressions (e.g., when both $k_{\mathcal{Y}}$ and $p_1, \ldots, p_n$ are Gaussian), then the conditional kernel mean is also given in closed form.

B Proof of Proposition 1

Proof We can expand the squared error in the RKHS $\mathcal{H}_{\mathcal{Y}}$ as

$$
\begin{aligned}
&\|\hat{m}_{Q_{\mathcal{Y}}} - m_{Q_{\mathcal{Y}}}\|^2_{\mathcal{H}_{\mathcal{Y}}} \\
&= \left\| \sum_{i=1}^{\ell} \gamma_i m_{\mathcal{Y}|\tilde{X}_i} - m_{Q_{\mathcal{Y}}} \right\|^2_{\mathcal{H}_{\mathcal{Y}}} \\
&= \sum_{i,j=1}^{\ell} \gamma_i \gamma_j \left\langle m_{\mathcal{Y}|\tilde{X}_i}, m_{\mathcal{Y}|\tilde{X}_j} \right\rangle_{\mathcal{H}_{\mathcal{Y}}} - 2 \sum_{i=1}^{\ell} \gamma_i \left\langle m_{\mathcal{Y}|\tilde{X}_i}, m_{Q_{\mathcal{Y}}} \right\rangle_{\mathcal{H}_{\mathcal{Y}}} + \left\| m_{Q_{\mathcal{Y}}} \right\|^2_{\mathcal{H}_{\mathcal{Y}}} \\
&= \sum_{i,j=1}^{\ell} \gamma_i \gamma_j \int\int k_{\mathcal{Y}}(y, \tilde{y}) p_M(y|\tilde{X}_i) p_M(\tilde{y}|\tilde{X}_j) dy d\tilde{y} \\
&\quad -2 \sum_{i=1}^{\ell} \gamma_i \int \left(\int\int k_{\mathcal{Y}}(y, \tilde{y}) p_M(y|\tilde{X}_i) p_M(\tilde{y}|x) dy d\tilde{y} \right) \pi(x) dx \\
&\quad + \int\int \left(\int\int k_{\mathcal{Y}}(y, \tilde{y}) p_M(y|x) p_M(\tilde{y}|\tilde{x}) dy d\tilde{y} \right) \pi(x) \pi(\tilde{x}) dx d\tilde{x} \\
&= \sum_{i,j=1}^{\ell} \gamma_i \gamma_j \theta(\tilde{X}_i, \tilde{X}_j) - 2 \sum_{i=1}^{\ell} \gamma_i \int \theta(\tilde{X}_i, x) \pi(x) dx + \int\int \theta(x, \tilde{x}) \pi(x) \pi(\tilde{x}) dx d\tilde{x} \\
&= \left\langle (\hat{m}_\Pi - m_\Pi) \otimes (\hat{m}_\Pi - m_\Pi), \theta \right\rangle_{\mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}} \\
&\leq \left\| \hat{m}_\Pi - m_\Pi \right\|^2_{\mathcal{H}_{\mathcal{X}}} \|\theta\|_{\mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}},
\end{aligned}
$$

where the fifth equality follows from the assumption that $\theta \in \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}$. The assertion then follows from $||\hat{m}_\Pi - m_\Pi||_{\mathcal{H}_{\mathcal{X}}} = O_p(\ell^{-\alpha})$ and $\|\theta\|_{\mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}} < \infty$. □

References

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, *68*(3), 337–404.

Boots, B., Gordon, G., & Gretton A. (2013). Hilbert space embeddings of predictive state representations. In *The conference on uncertainty in artificial intelligence (UAI)* (pp. 92–101).

Briol, F., Oates, C. J., Girolami, M., Osborne, M. A., & Sejdinovic, D. (2019). Probabilistic integration: A role in statistical computation? *Statistical Science*, *34*(1), 1–22.

Caponnetto, A., & Vito, E. D. (2007). Optimal rates for regularized least-squares algorithm. *Found Comput Math J*, *7*(4), 331–368.

Chen, Y., Welling, M., & Smola, A. (2010). Super-samples from kernel herding. In *Proceedings of the twenty-sixth conference on uncertainty in artificial intelligence* (pp 109–116). AUAI Press.

Cockayne, J., Oates, C., Sullivan, T., & Girolami, M. (2019). Bayesian probabilistic numerical methods. *SIAM Review* (to appear).

Cortes, C., Mohri, M., & Weston, J. (2005). A general regression technique for learning transductions. In *International conference on machine learning (ICML)* (pp. 153–160).

Deisenroth, M. P., Huber, M. F., & Hanebeck, U. D. (2009). Analytic moment-based Gaussian process filtering. In *International conference on machine learning (ICML)* (pp. 225–232).
Deisenroth, M., Turner, R., Huber, M., Hanebeck, U., & Rasmussen, C. (2012). Robust filtering and smoothing with Gaussian processes. *IEEE Transactions on Automatic Control*, *57*(7), 1865–1871.
Doucet, A., Freitas, N. D., & Gordon, N. J. (Eds.) (2001). *Sequential Monte Carlo methods in practice*. Berlin: Springer.
Doucet, A., & Johansen, A. M. (2011). A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan & B. Rozovskii (Eds.), *The Oxford handbook of nonlinear filtering* (pp. 656–704). Oxford: Oxford University Press.
Evensen, G. (2009). *Data assimilation: The ensemble Kalman filter*. Berlin: Springer.
Fukumizu, K., Bach, F. R., & Jordan, M. I. (2004). Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research*, *5*, 73–99.
Fukumizu, K., Song, L., & Gretton, A. (2013). Kernel Bayes' rule: Bayesian inference with positive definite kernels. *Journal of Machine Learning Research*, *14*, 3753–3783.
Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. J. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, *13*, 723–773.
Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. (2005). Measuring statistical dependence with Hilbert–Schmidt norms. In S. Jain, H. U. Simon, & E. Tomita (Eds.), *Algorithmic learning theory* (Vol. 3734, pp. 63–77)., Lecture notes in computer science Berlin: Springer.
Grünewälder, S., Lever, G., Baldassarre, L., Patterson, S., Gretton, A., & Pontil, M. (2012a). Conditional mean embeddings as regressors—supplementary. In *International conference on machine learning (ICML)* (pp 1823–1830).
Grünewälder, S., Lever, G., Baldassarre, L., Pontil, M., & Gretton, A. (2012b) Modelling transition dynamics in MDPs with RKHS embeddings. In *International conference on machine learning (ICML)* (pp. 535–542).
Hammerstein, E. A. F. V. (2010). *Generalized hyperbolic distributions: Theory and applications to CDO pricing*. Ph.D. thesis, University of Freiburg.
Hennig, P., Osborne, M. A., & Girolami, M. (2015). Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, *471*, 20150142. https://doi.org/10.1098/rspa.2015.0142.
Hsu, K., & Ramos, F. (2019). Bayesian learning of conditional kernel mean embeddings for automatic likelihood-free inference. In *Proceedings of the 22nd international conference on artificial intelligence and statistics (AISTATS 2019), PMLR* (Vol. 89, pp. 2631–2640). http://proceedings.mlr.press/v89/hsu19a.html.
Julier, S. J., & Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *IEEE Review*, *92*, 401–422.
Kajihara, T., Kanagawa, M., Yamazaki, K., & Fukumizu, K. (2018). Kernel recursive ABC: Point estimation with intractable likelihood. In: J. Dy, & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning, PMLR, Stockholmsmässan, Stockholm Sweden, Proceedings of machine learning research* (Vol. 80, pp. 2400–2409). http://proceedings.mlr.press/v80/kajihara18a.html.
Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, *82*, 35–45.
Kanagawa, M., Hennig, P., Sejdinovic, D., & Sriperumbudur, B. K. (2018). Gaussian processes and kernel methods: A review on connections and equivalences. arXiv: arXiv:1805.08845v1 [stat.ML].
Kanagawa, M., Nishiyama, Y., Gretton, A., & Fukumizu, K. (2016a). Filtering with state-observation examples via kernel Monte Carlo filter. *Neural Computation*, *28*, 382–444.
Kanagawa, M., Sriperumbudur, B. K., & Fukumizu, K. (2016b). Convergence guarantees for kernel-based quadrature rules in misspecified settings. In *Neural information processing systems (NIPS)* (pp. 3288–3296).
Kanagawa, M., Sriperumbudur, B. K., & Fukumizu, K. (2019). Convergence analysis of deterministic kernel-based quadrature rules in misspecified settings. *Foundations of Computational Mathematics*. https://doi.org/10.1007/s10208-018-09407-7 (to appear).
Kersting, H., & Hennig, P. (2016). Active uncertainty calibration in Bayesian ode solvers. In *Proceedings of the 32nd conference on uncertainty in artificial intelligence (UAI 2016)* (pp. 309–318). AUAI Press. http://www.auai.org/uai2016/proceedings/papers/163.pdf.
Ko, J., & Fox, D. (2009). GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models. *Auton Robots*, *27*(1), 75–90.
Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *The IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2169–2178).
Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Matèrn, B. (1986). *Spatial variation* (2nd ed.). Berlin: Springer.
McCalman, L. (2013). *Function embeddings for multi-modal Bayesian inference*. A Ph.D. thesis in the University of Sydney. http://hdl.handle.net/2123/12031.
McCalman, L., O'Callaghan, S., & Ramos, F. (2013). Multi-modal estimation with kernel embeddings for learning motion models. In *IEEE international conference on robots and automation (ICRA)*.
Mika, S., Schölkopf, B., Smola, A., Müller, K., Scholz, M., & Rätsch, G. (1999). Kernel PCA and de-noising in feature spaces. In *Neural information processing systems (NIPS)* (pp 536–542).
Mitrovic, J., Sejdinovic, D., & Teh, Y. W. (2016). Dr-abc: Approximate Bayesian computation with kernel-based distribution regression. In: M. F. Balcan, K. Q. Weinberger (Eds.), *Proceedings of the 33rd international conference on machine learning*, PMLR, New York, New York, USA, *Proceedings of machine learning research* (Vol. 48, pp. 1482–1491).
Morere, P., Marchant, R., & Ramos, F. (2018). Continuous state-action-observation POMDPs for trajectory planning with Bayesian optimisation. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 8779–8786). https://doi.org/10.1109/IROS.2018.8593850.
Muandet, K., Fukumizu, K., Sriperumbudur, B., & Schölkopf, B. (2017). Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends in Machine Learning*, *10*(1–2), 1–141.
Nakagome, S., Fukumizu, K., & Mano, S. (2013). Kernel approximate Bayesian computation in population genetic inferences. *Statistical Applications in Genetics and Molecular Biology*, *12*(6), 667–678.
Nishiyama, Y., & Fukumizu, K. (2016). Characteristic kernels and infinitely divisible distributions. *Journal of Machine Learning Research*, *17*(180), 1–28.
Nishiyama, Y., Boularias, A., Gretton, A., & Fukumizu, K. (2012). Hilbert space embeddings of POMDPs. In *The conference on uncertainty in artificial intelligence (UAI)* (pp. 644–653).
Nishiyama, Y., Afsharinejad, A. H., Naruse, S., Boots, B., & Song, L. (2016). The nonparametric kernel Bayes' smoother. In *International conference on artificial intelligence and statistics (AISTATS)* (pp. 547–555).
Oates, C. J., & Sullivan, T. J. (2019). A modern retrospective on probabilistic numerics. *Statistics and Computing* (to appear).
Pronobis, A., & Caputo, B. (2009). COLD: COsy localization database. *The International Journal of Robotics Research (IJRR)* 28(5):588–594. Copyright 2009 by the Authors. Reprinted by permission of SAGE Publications, Ltd.
Rasmussen, C., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Cambridge, MA: MIT Press.
Rawlik, K., Toussaint, M., & Vijayakumar, S. (2013). Path integral control by reproducing kernel Hilbert space embedding. In *Proceedings of the 23rd international joint conference on artificial intelligence (IJCAI)*.
Saito, T. (2019). *Tsunami generation and propagation*. Berlin: Springer.
Schober, M., Duvenaud, D., & Hennig, P. (2014). Probabilistic ODE solvers with Runge–Kutta means. In *Advances in neural information processing systems 27* (pp. 739–747), Curran Associates, Inc. http://papers.nips.cc/paper/5451-probabilistic-ode-solvers-with-runge-kutta-means.pdf.
Schober, M., Särkkä, S., & Hennig, P. (2018). A probabilistic model for the numerical solution of initial value problems. *Statistics and Computing*,. https://doi.org/10.1007/s11222-017-9798-7.
Schölkopf, B., & Smola, A. (2002). *Learning with kernels*. Cambridge: MIT Press.
Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. London: Chapman and Hall.
Simon-Gabriel, C. J., Schölkopf, B. (2018). Kernel distribution embeddings: Universal kernels, characteristic kernels and kernel metrics on distributions. *Journal of Machine Learning Research*, *19*(44), 1–29. http://jmlr.org/papers/v19/16-291.html.
Smola, A., Gretton, A., Song, L., & Schölkopf, B. (2007). A Hilbert space embedding for distributions. In *International conference on algorithmic learning theory (ALT)* (pp. 13–31).
Song, L., Huang, J., Smola, A., & Fukumizu, K. (2009). Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *International conference on machine learning (ICML)* (pp. 961–968).
Song, L., Gretton, A., Bickson, D., Low, Y., & Guestrin, C. (2011). Kernel belief propagation. *Journal of Machine Learning Research—Proceedings Track*, *15*, 707–715.
Song, L., Fukumizu, K., & Gretton, A. (2013). Kernel embedding of conditional distributions. *IEEE Signal Processing Magazine*, *30*(4), 98–111.
Sriperumbudur, B., Gretton, A., Fukumizu, K., Lanckriet, G., & Schölkopf, B. (2010). Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, *11*, 1517–1561.
Sriperumbudur, B., Fukumizu, K., Gretton, A., Schölkopf, B., & Lanckriet, G. (2012). On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, *6*, 1550–1599.
Steinwart, I., & Christmann, A. (2008). *Support vector machines. Information science and statistics*. Berlin: Springer.

Sudderth, E. B., Ihler, A. T., Isard, M., Freeman, W. T., & Willsky, A. S. (2010). Nonparametric belief propagation. *Communications of the ACM*, *53*(10), 95–103.
Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. Cambridge, MA: MIT Press.
Tolstikhin, I., Sriperumbudur, B. K., & Muandet, K. (2017). Minimax estimation of kernel mean embeddings. *Journal of Machine Learning Research 18*(86), 1–47. http://jmlr.org/papers/v18/17-032.html.
Tronarp, F., Kersting, H., Särkkä, S., & Hennig, P. (2018). Probabilistic solutions to ordinary differential equations as non-linear Bayesian filtering: A new perspective. ArXiv preprint arXiv:1807.09737 [stat.ME].
Vlassis, N., Terwijn, B., & Kröwe, B. (2002). Auxiliary particle filter robot localization from high-dimensional sensor observations. In *Proceedings of the international conference on robotics and automation (ICRA)* (pp 7–12).
Weston, J., Chapelle, O., Elisseeff, A., Schölkopf, B., & Vapnik, V. (2003). Kernel dependency estimation. *Advances in Neural Information Processing Systems*, *15*, 873–880.
Winsberg, E. (2010). *Science in the age of computer simulation*. Chicago: University of Chicago Press.

Affiliations

Yu Nishiyama[1] · Motonobu Kanagawa[2] · Arthur Gretton[3] · Kenji Fukumizu[4]

✉ Yu Nishiyama
ynishiyam@gmail.com
https://sites.google.com/site/ynishiyam/

Motonobu Kanagawa
motonobu.kanagawa@gmail.com
https://sites.google.com/site/motonobukanagawa/

Arthur Gretton
arthur.gretton@gmail.com
http://www.gatsby.ucl.ac.uk/∼gretton/

Kenji Fukumizu
fukumizu@ism.ac.jp
https://www.ism.ac.jp/∼fukumizu/

1 The University of Electro-Communications, Chofu, Japan

2 University of Tübingen, Tübingen, Germany

3 Gatsby Computational Neuroscience Unit, University College London, London, England

4 The Institute of Statistical Mathematics, Tachikawa, Japan