# Empirical representations of probability distributions via kernel mean embeddings

Motonobu Kanagawa

Doctor of Philosophy

Department of Statistical Science
School of Multidisciplinary Sciences
SOKENDAI (The Graduate University for Advanced Studies)

# Empirical representations of probability distributions via kernel mean embeddings

A DISSERTATION
SUBMITTED TO THE FACULTY OF
THE SCHOOL OF MULTIDISCIPLINARY SCIENCES
THE DEPARTMENT OF STATISTICAL SCIENCE
THE GRADUATE UNIVERSITY FOR ADVANCED STUDIES
BY

Motonobu Kanagawa

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Kenji Fukumizu, Advisor

March 2016

# Acknowledgements

I am grateful to all people whose have supported me in the process of my Ph.D. course. First and foremost, I would like to express my gratitude to Prof. Kenji Fukumizu for being my supervisor since I came to the Institute of Statistical Mathematics as an intern. I would like to thank Prof. Satoshi Kuriki, Prof. Daichi Mochihashi, Prof. Ryo Yoshida, and Prof. Taiji Suzuki for being committee members and for their helpful comments, and Prof. Arthur Gretton and Prof. Yu Nishiyama for being collaborators on the works that become a basis of this thesis. I am also grateful to the members of the Fukumizu laboratory and the SML center, and students and professors of ISM.

Finally, I would like to thank my family for their warm supports.

# Abstract

How to represent probability distributions is a fundamental issue in statistical methods, as this determines all the subsequent estimation procedures. In this thesis, we focus on the approach using positive definite kernels and reproducing kernel Hilbert spaces (RKHS), namely the framework of kernel mean embeddings. In this framework, any probability distribution is represented as an element in an RKHS, which is called the kernel mean. Since each kernel mean contains all the information about the embedded distribution, statistical inference can be conducted by estimating the kernel means of distributions which one is interested in. In general, a finite sample estimate of a kernel mean is given as a weighted sum of feature vectors. Therefore an empirical kernel mean is expressed by a weighted sample, and thus it is similar to an empirical distribution.

In this thesis, we investigate this similarity between empirical kernel means and empirical distributions, and show that empirical kernel means can be treated as empirical distributions; this enables us to incorporate Monte Carlo methods into the framework of kernel mean embeddings. We first prove that a sampling method can be applied to empirical kernel means, as is commonly done to empirical distributions in the field of Monte Carlo methods. Based on this theoretical result, we develop a novel method for filtering in a state-space model, which effectively combines the sampling method and a nonparametric learning approach of kernel mean embeddings. We also prove that empirical kernel means can be used for estimating expectations of functions of a broad class, similarly to empirical distributions.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Knowing distributions of samples is the ultimate goal of statistical science. This means that many statistical problems may be cast as the problem of estimating unknown distributions.

For instance, let us consider the two sample problem, where one is given independent samples from two distributions, and the task is to test the homogeneity of these distributions. This can be done by first estimating the two distributions, and then comparing the resulting distribution estimates. Similarly, the problem of measuring and testing dependency between two random variables can be seen as that of distribution estimation. This is because statistical dependency is measured by the discrepancy between the joint distribution of the two random variables and the product distribution of their marginals. Therefore the dependency can be estimated by estimating and comparing these two distributions.

Problems of prediction, rather than those of inference as above, can be also formulated as the estimation of unknown distributions. The simplest example is regression, which is essentially the task of estimating the conditional distribution of a response given covariates. Another typical example is the prediction of future observations in time-series data analysis; this is the problem of estimating the distribution of future observations.

## 1.1 Representations of probability distributions

What is the meaning of "the estimation of probability distributions"? Probability distributions are measures, so one might think of it as the estimation of measures themselves. However, the estimation of measures is not straightforward in practice and may be infeasible. Alternatively, we can consider distribution estimation as the problem of estimating some quantities which are uniquely associated to the distributions of interest; we will call such quantities *representations of probability distributions*

in this thesis. The following are examples what we consider as probability representations:

**Example 1** (Parameters of a parametric family of distributions)**.** *Suppose one is interested in distributions belonging to a parametric family of distributions* $\{P_\theta : \theta \in \Theta\}$ *indexed by finite dimensional vectors* $\theta$ *in a certain set* $\Theta$*. Since each parameter vector* $\theta$ *is uniquely associated to some distribution* $P_\theta$*, one can consider these vectors as representations of distributions in the parametric family. Therefore estimating a parameter vector* $\theta$ *amounts to estimating the corresponding distribution* $P_\theta$*.*

**Example 2** (Density functions in nonparametric models)**.** *Suppose one is interested in a class of distributions that admit density functions of a certain degree of smoothness, i.e., a nonparametric density model. In this case, the density functions can be seen as representations of the distributions, since they identify the corresponding distributions in the model.*

**Example 3** (Characteristic functions)**.** *Suppose one is interested in all probability distributions on the Euclidian space* $\mathbb{R}^d$*. For any distribution* $P$*, the characteristic function is defined as the (inverse) Fourier transform of* $P$*:*

$$\phi_P(w) := \int_{\mathbb{R}^d} e^{\sqrt{-1}x^T w} dP(x), \quad w \in \mathbb{R}^d. \tag{1.1}$$

*Characteristic functions and distributions are one-to-one (Dudley, 2002, Section 9.5), so one can think of characteristic functions as representations of all distributions on* $\mathbb{R}^d$*.*

For example, consider again the two-sample problem, and suppose there exist some representations of the two distributions. Then testing whether these representations are equal or not yields a solution of the two sample problem, since they are uniquely associated to the two distributions under consideration. Therefore it suffices to estimate those quantities that represent the two distributions.

To make use of such representations in practice, however, one must consider how to estimate them given samples. For parametric models, there are several methods for parameter estimation, ranging from maximum likelihood estimation to Bayesian methods. On the other hand, density functions can be estimated nonparametrically, using smoothing methods such as kernel density estimation (Silverman, 1986). Characteristic functions may be estimated as empirical averages of the exponential functions (Feuerverger and Mureika, 1977).

In the estimation of probability representations, there is a tradeoff between the richness of representations and convergence rates. This is the so-called bias-variance tradeoff. For example, parametric models only cover distributions of a finite number

of degrees of freedom, so they are more restrictive than nonparametric models. However, convergence rates are fast and independent of the dimensionality of data. On the other hand, nonparametric models can deal with a wider class of distributions than parametric models, but convergence rates are slow due to the so-called curse of dimensionality: to achieve a certain level of accuracy, the number of samples needs to be exponential to the dimensionality of data (Silverman, 1986).

Characteristic functions enjoy the merits of both parametric and nonparametric approaches: (a) they can represent all distributions on $\mathbb{R}^d$; (b) estimation with an i.i.d. sample can achieve the same convergence rate as those of parametric models (Feuerverger and Mureika, 1977). Nonetheless, the use of empirical characteristic functions in statistical inference and prediction may not be straightforward. For example, comparison of two characteristic functions may be possible by defining distance between them, such as the $L_2$ distance. However, the $L_2$ distance between characteristic functions does not allow an analytic formula with respect to samples, so the computation might require numerical integration[1] (Yu, 2004).

## 1.2 Kernel mean embeddings of distributions

The focus of this thesis is on probability representations based on positive definite kernels and associated reproducing kernel Hilbert spaces (RKHS) (Berlinet and Thomas-Agnan, 2004, Chapter 4). This framework has been developed in the machine learning community in the last decades (Smola et al., 2007; Sriperumbudur et al., 2010; Song et al., 2013).

Let $\mathcal{X}$ be a measurable space, $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel and $\mathcal{H}$ be the RKHS associated with $k$ (definitions will be given in Chapter 2). The RKHS is a function space consisting of functions on $\mathcal{X}$, properties of which are determined by the kernel $k$. Then for any probability distribution $P$ on $\mathcal{X}$, its representation in the RKHS $\mathcal{H}$ is defined as the expectation of the kernel function with respect to that distribution:

$$m_P := \mathbf{E}_{X \sim P}[k(\cdot, X)] = \int k(\cdot, x) dP(x). \tag{1.2}$$

This is called the *kernel mean* of the distribution $P$. For a rich enough RKHS, it can be shown that probability distributions and kernel means is one-to-one (Sriperumbudur et al., 2010): for any distributions $P$ and $Q$, $m_P = m_Q$ holds if and only if $P = Q$. Therefore kernel means are valid as representations of probability distributions.

[1] While the $L_2$ distance between empirical characteristic functions may not have an analytic solution, a certain *weighted* $L_2$ distance does; the resulting distance is known as the energy distance (Székely and Rizzo, 2013, Proposition 1). The energy distance has a close relationship with kernel mean embeddings discussed in this thesis, as shown by Sejdinovic et al. (2013b).

This approach is akin to representations given by characteristic functions in the following points: (i) the representation of a distribution is given as the expectation of a certain function (i.e. kernel or exponential function) with respect to that distribution; (ii) distribution representations and distributions are one-to-one. In fact, there is a close relationship between these two approaches; one can define a metric on distributions as the RKHS distance between kernel means, and this metric may be written as a weighted $L_2$ distance between characteristic functions (Sriperumbudur et al., 2010).

Compared to existing nonparametric approaches based on densities or characteristic functions, however, kernel mean embeddings have the following advantages:

1. Computation of empirical quantities often allow analytic solutions via simple linear algebraic operations. For instance, the RKHS distance between empirical kernel means can be computed by evaluation of kernel values between samples, which results in computation of kernel matrices. This comes from the so-called the reproducing property of the RKHS.

2. As in other kernel methods in machine learning (Schölkopf and Smola, 2002), empirical performance is not strongly affected by the superficial dimensionality of data. For instance, given an i.i.d. sample of size $n$ from a distribution, the corresponding kernel mean is estimated at a convergence rate $O_p(n^{-1/2})$, which is independent of the dimensionality.

3. The domain of data can be arbitrary, as long as positive definite kernels are defined on that space (e.g. structured data such as images, texts, and graphs).

Because of these merits, kernel mean embeddings have been used in a variety of problems in statistics and machine learning. For instance, applications include hypothesis testing such as two sample and independence testing (Gretton et al., 2012, 2008; Sejdinovic et al., 2013a), measures of dependency between random variables (Fukumizu et al., 2004, 2009a; Gretton et al., 2005; Fukumizu et al., 2008), state-space models (Song et al., 2009, 2010a; Fukumizu et al., 2013; Zhu et al., 2014; Kanagawa et al., 2014), belief propagation (Song et al., 2010b, 2011a), graphical model learning (Song et al., 2011b; Song and Dai, 2013; Song et al., 2014), predictive state representations (Boots et al., 2013, 2014), and reinforcement learning (Grünewälder et al., 2012b; Nishiyama et al., 2012; van Hoof et al., 2015).

In these applications, a kernel mean $m_P$ is estimated as a weighted sum of kernel functions with some weights $w_1, \ldots, w_n \in \mathbb{R}$ and samples $X_1, \ldots, X_n \in \mathcal{X}$:

$$\hat{m}_P := \sum_{i=1}^{n} w_i k(\cdot, X_i), \tag{1.3}$$

These weights are typically obtained via linear algebraic operations on kernel matrices, such as regularized matrix inversion (Song et al., 2009, 2013; Fukumizu et al., 2013), singular value decomposition (Song et al., 2010a, 2011b), tensor decomposition (Song et al., 2014), or their combinations. The weights are computed so as to make the estimate (1.3) close to the kernel mean $m_P$, so they can take negative values, and their sum is not necessarily 1.

## 1.3 Empirical distributions and Monte Carlo methods

In this thesis, we are interested in the similarity of empirical kernel means (1.3) and *empirical distributions*. An empirical distribution of a probability distribution $P$ is given as

$$\hat{P} = \sum_{i=1}^{n} w_i \delta_{X_i} \tag{1.4}$$

with some samples $X_1, \ldots, X_n \in \mathcal{X}$ and weights $w_1, \ldots, w_n \geq 0$, where $\delta_x$ denotes the Dirac measure at $x$. It is an approximation of the distribution $P$ in that it provides approximations of function value expectations with respect to $P$ (Douc and Moulines, 2008). Namely, let $\mathcal{F}$ be a certain set of test functions (e.g. the set of all bounded continuous functions). Then for any $f \in \mathcal{F}$, the weighted average $\sum_{i=1}^{n} w_i f(X_i)$ should be a good approximation of the expectation $\mathbf{E}_{X \sim P}[f(X)]$.

Empirical distributions are basis of Monte Carlo methods, as weighted samples are useful for manipulating distributions by simulation. For example, in sequential Monte Carlo (SMC) methods (Doucet et al., 2001), forward probabilities are computed by simulating samples and propagating the associated weights. In Markov chain Monte Carlo (MCMC) methods (Robert and Casella, 2004), new samples are generated conditioned on previously generated samples. The focus of the Monte Carlo methods is not on the estimation unknown distributions: distributions are assumed known in a certain sense; in standard Monte Carlo integration, distributions are known entirely, while in MCMC, distributions are known up to their normalization constants. These methods aims to numerically approximate an intractable integral

$$\int f(x) dP(x) \tag{1.5}$$

for a test function $f$ of a certain class (e.g. polynomials that yield moments). This is done by generating weighted samples $\{(w_i, X_i)\}$, and computing the weighted sum of

function values:

$$\sum_{i=1}^{n} w_i f(X_i). \tag{1.6}$$

How to generate the weighted samples depend on the method employed; for the standard Monte Carlo and MCMC, the weights are defined as uniform, while for importance sampling and SMC, non-uniform weights are used as each weight represents importance of the associated sample (Liu, 2001).

As mentioned earlier, empirical kernel means (1.3) are similar to empirical distributions, as both of them are represented by weighted samples $\{(w_i, X_i)\}$; the difference is that kernels are used in place of Dirac measures, and the weights may take negative values. Recall that (1.3) is an approximation of the kernel mean $m_P$. From this, it can be easily shown (see Chapter 2) that the weighted sum $\sum_{i=1}^{n} w_i f(X_i)$ for any function $f$ in the RKHS will be a good approximation of the expectation $\mathbf{E}_{X\sim P}[f(X)]$ with respect to $P$. In this sense, the empirical kernel mean can also be seen as an empirical distribution, with test functions being those in the RKHS.

## 1.4 Contributions

The aim of this thesis is to introduce the following perspective to the theory of kernel mean embeddings: *empirical kernel means can be treated as empirical distributions.* This is motivated by the similarity between empirical kernel means and empirical distributions mentioned above. The further investigation of this similarity is important, as it implies that Monte Carlo methods may be combined with empirical kernel means.

Specifically, we ask the following questions, which we address in the chapters written in parentheses:

1. Is it possible to treat empirical kernel means as if they were empirical distributions? More specifically, can we apply operations of Monte Carlo methods to empirical kernel means, as for empirical distributions? (Chapter 3)

2. Can we combine techniques of Monte Carlo methods with learning and inference methods based on kernel mean embeddings? (Chapter 4)

3. Are the test functions for empirical kernel means only restricted to those in the RKHS? In other words, can we estimate expectations of functions outside the RKHS, in the same way as for those in the RKHS? (Chapter 5)

Below we explain these questions along with the contributions of this thesis. After reviewing the theory of kernel mean embeddings in Chapter 2, this thesis proceeds as follows.

**Chapter 3.** This chapter is devoted to theoretical analysis, and discusses applicability of a sampling method with an empirical kernel mean. Specifically, we consider a sampling procedure using a conditional distribution, which corresponds to that of a particle filter for computing forward probabilities. We prove that such a sampling method can in fact be used with an empirical kernel mean. More precisely, we prove that this sampling method yields a consistent kernel mean estimator for a forward probability, if the given empirical kernel mean is consistent.

Mainly there are three basic operations on empirical distributions used in Monte Carlo or particle methods (Liu, 2001; Doucet et al., 2001; Doucet and Johansen, 2011):

1. Sampling: generating samples from a conditional distribution, conditioned on samples of an empirical distribution.

2. Importance weighting: updating the weights, by multiplying the importance of each sample to the associated weight.

3. Resampling: sampling from the empirical distribution without replacement, by regarding it as a discrete distribution.

Our analysis reveals that the first operation (sampling) can also be employed with empirical kernel means: this makes it possible to combine the sampling method with existing learning and inference methods of kernel mean embeddings. On the other hand, the other operations are not straightforward to realize with kernel mean embeddings. This is because these operations make use of the positiveness of the weights of empirical distributions, while those of empirical kernel means can be negative. Our analysis also reveals that resampling can be beneficial to improve the accuracy of the sampling procedure, so it would be desirable to realize it with kernel mean embeddings. To this end, we propose a novel resampling algorithm based on the Kerne Herding algorithm by Chen et al. (2010). We also provide detailed theoretical analysis of this method, and explain its mechanism.

**Chapter 4.** In this chapter, we demonstrate how the above procedures can be combined with existing methods of kernel mean embeddings. Specifically, we develop a novel filtering method for state-space models, which we call *Kernel Monte Carlo Filter* (KMCF).

The proposed method is a combination of the sampling and resampling methods in Chapter 3 and Kernel Bayes Rule by Fukumizu et al. (2013). This filtering method focuses on the setting where the observation model is to be learned from state-observation examples, while the state-transition model is known and sampling is possible. We make use of the sampling and resampling procedures to handle the

state-transition model, while using the Kernel Bayes' Rule to learn the observation model.

This setting is useful in applications where the state variable are defined quantities that are very different from observations. We demonstrate our method in synthetic and real data experiments, which include the challenging problem of vision-based robot localization in robotics.

**Chapter 5.** In this chapter, we conduct theoretical analysis to investigate whether the weighted sum (1.6) becomes a consistent estimator of the function value expectation (1.5) when the test function does not belong to the RKHS. This question is motivated by conceptual and practical reasons. Conceptually, a consistent kernel mean estimator should provide all the information about the distribution $P$, since the kernel mean $m_P$ uniquely identifies this distribution. The practical reason is that RKHSs of widely used kernels (e.g. Gaussian) often do not contain important functions for statistical inference. For example, polynomial functions and indicator functions are not contained in the Gaussian RKHS: expectations of these provide moments and confidence intervals, respectively. This means that it is not guaranteed whether these quantities can be estimated by a consistent kernel mean estimator.

By technical reasons, we focus on kernel mean embeddings using the Gaussian kernel and its RKHS. We prove that in this case, expectations of functions in the *Besov space* can be estimated with a consistent kernel mean estimator. The Besov space is a generalization of the Sobolev space, and consists of functions with a certain degree of smoothness. It contains functions which are less smooth than those in the Gaussian RKHS. Therefore our results guarantee that the weighted sum (1.6) can be consistent for functions having a certain degree of smoothness, even when these functions do not belong to the Gaussian RKHS. As a corollary, we show that the moments and probably masses on cubes can be estimated with a consistent kernel mean estimator. This result is practically important, as it shows that these important quantities can in fact be estimated with kernel mean embeddings. Finally, we also show that the density can be estimated from a consistent kernel mean estimator. This result is useful in applications where the information of densities is important (e.g., MAP estimation in Bayesian inference).

Chapter 3 and Chapter 4 are based on the following journal and conference papers:

- Kanagawa, M., Nishiyama, Y., Gretton, A., and Fukumizu, K. (2016). Filtering with State-Observation Examples via Kernel Monte Carlo Filter. *Neural Computation*, volume 28, issue 2, pages 382–444.

- Kanagawa, M., Nishiyama, Y., Gretton, A., and Fukumizu, K. (2014). Monte Carlo filtering using kernel embedding of distributions. In *Proceedings of the*

*28th AAAI Conference on Artificial Intelligence (AAAI-14)*, pages 1897–1903.

Chapter 5 is based on the following conference paper:

- Kanagawa, M. and Fukumizu, K. (2014). Recovering distributions from Gaussian RKHS embeddings. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS 2014)*, pages 457–465.

# Chapter 2

# Kernel mean embeddings of distributions

In this chapter, we review the framework of kernel mean embeddings.

## 2.1 Positive definite kernels and reproducing kernel Hilbert spaces

We begin by introducing positive definite kernels and reproducing kernel Hilbert spaces (RKHS), details of which can be found in Schölkopf and Smola (2002); Berlinet and Thomas-Agnan (2004); Steinwart and Christmann (2008).

Let $\mathcal{X}$ be a set, and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite (p.d.) kernel: a symmetric kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called positive definite, if for all $n \in \mathbb{N}$, $c_1, \ldots, c_n \in \mathbb{R}$, and $X_1, \ldots, X_n \in \mathcal{X}$, we have

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(X_i, X_j) \geq 0.$$

Any positive definite kernel is uniquely associated with a Reproducing Kernel Hilbert Space (RKHS) (Aronszajn, 1950). Let $\mathcal{H}$ be the RKHS associated with $k$. The RKHS $\mathcal{H}$ is a Hilbert space of functions on $\mathcal{X}$, which satisfies the following important properties:

1. (**feature vector**): $k(\cdot, x) \in \mathcal{H}$ for all $x \in \mathcal{X}$.
2. (**reproducing property**): $f(x) = \langle f, k(\cdot, x) \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$ and $x \in \mathcal{X}$,

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the inner product equipped with $\mathcal{H}$, and $k(\cdot, x)$ is a function with $x$ fixed.

The reproducing property is why the Hilbert $\mathcal{H}$ is called the reproducing kernel Hilbert space. Combined with the first property, it implies

$$k(x, x') = \langle k(\cdot, x), k(\cdot, x') \rangle_{\mathcal{H}}, \quad \forall x, x' \in \mathcal{X}. \tag{2.1}$$

Namely, $k(x, x')$ implicitly computes the inner product between the functions $k(\cdot, x)$ and $k(\cdot, x')$. Therefore $k(\cdot, x)$ can be seen as an implicit representation of $x$ in $\mathcal{H}$. In fact, the RKHS is often high-dimensional (or even infinite dimensional), and thus the function $k(\cdot, x)$ provides a high-dimensional representation of the data. Therefore $k(\cdot, x)$ is called the *feature vector* of $x$, and $\mathcal{H}$ the feature space.

It is also well-known (Aronszajn, 1950) that the subspace spanned by the feature vectors $\{k(\cdot, x) | x \in \mathcal{X}\}$ is dense in $\mathcal{H}$. This means that any function $f$ in $\mathcal{H}$ can be written as the limit of functions of the form $f_n := \sum_{i=1}^{n} c_i k(\cdot, X_i)$, where $c_1, \ldots, c_n \in \mathbb{R}$ and $X_1, \ldots, X_n \in \mathcal{X}$.

For example, positive definite kernels on the Euclidian space $\mathcal{X} = \mathbb{R}^d$ include Gaussian kernel $k(x, x') = \exp(-\|x - x'\|_2^2 / 2\sigma^2)$ and Laplace kernel $k(x, x') = \exp(-\|x - x\|_1 / \sigma)$, where $\sigma > 0$ and $\|\cdot\|_1$ denotes the $\ell_1$ norm. Notably, kernel methods allow $\mathcal{X}$ to be a set of *structured data*, such as images, texts or graphs. In fact, there exist various positive definite kernels developed for such structured data (Hofmann et al., 2008). Note that the notion of positive definite kernels is *different* from smoothing kernels in kernel density estimation (Silverman, 1986): a smoothing kernel does not necessarily define an RKHS.

In machine learning, positive definite kernels and RKHSs have been widely used for constructing nonlinear learning methods from the corresponding linear ones (Schölkopf and Smola, 2002; Hofmann et al., 2008). This can be done by representing each data $x \in \mathcal{X}$ as a feature vector $k(\cdot, x)$ in the RKHS $\mathcal{H}$, and defining a linear method in this RKHS. Then the resulting learning methods will be nonlinear to the original data. Significantly, such feature vectors, which can be infinite dimensional, need never be computed explicitly. This is because (i) the constructed linear method in the RKHS can be written in terms of the inner-product between feature vectors, and (ii) such inner-products can be computed by just evaluating kernel values between samples, thanks to the property (2.1). Popular examples of nonlinear methods constructed in this way include support vector machines (Vapnik, 1998; Steinwart and Christmann, 2008), kernel PCA (Schölkopf et al., 1998), and kernel CCA (Akaho, 2001; Bach and Jordan, 2002), among others; see also Schölkopf and Smola (2002); Hofmann et al. (2008).

## 2.2 Kernel means

We now show how to represent probability distributions using positive definite kernels and the associated RKHSs. Let $\mathcal{X}$ be a measurable space, $k$ be a measurable kernel on $\mathcal{X}$ that is bounded: $\sup_{x \in \mathcal{X}} k(x, x) < \infty$, and $\mathcal{H}$ be the RKHS of $k$. Let $P$ be an arbitrary probability distribution on $\mathcal{X}$. Then the representation of $P$ in the RKHS is defined as the mean of the feature vector:

$$m_P := \int k(\cdot, x) dP(x) \in \mathcal{H}, \tag{2.2}$$

which is called the **kernel mean** of $P$. This is a natural generalization of feature vector representations of individual points to probability distributions (Berlinet and Thomas-Agnan, 2004, Chapter 4). In fact, if the distribution $P$ is the Dirac measure $\delta_x$ at a point $x \in \mathcal{X}$, then the kernel mean $m_P$ becomes the feature vector $k(\cdot, x)$.

Is the kernel mean $m_P$ uniquely associated with the distribution $P$? In other words, does the kernel mean preserve all information of the embedded distribution? This question is very important for kernel means to be valid representations of distributions. This holds if the kernel $k$ is *characteristic*: a positive definite kernel $k$ is defined to be characteristic, if the mapping

$$P \to m_P \in \mathcal{H}$$

is injective (Fukumizu et al., 2004, 2008; Sriperumbudur et al., 2010). This means that the RKHS $\mathcal{H}$ is rich enough to distinguish among all distributions. For example, the Gaussian and Laplace kernels defined on $\mathbb{R}^d$ are characteristic. We discuss conditions for kernels being characteristic in Section 2.3.

An important property of the kernel mean (2.2) is the following: by the reproducing property, we have

$$\langle m_P, f \rangle_{\mathcal{H}} = \int f(x) dP(x) = \mathbf{E}_{X \sim P}[f(X)], \quad \forall f \in \mathcal{H}. \tag{2.3}$$

That is, the expectation of any function in the RKHS can be given by the inner product between the kernel mean and that function.

We can construct learning methods for kernel means, as have been done for feature vectors in standard kernel methods. This results in learning methods on probability distributions (i.e., each input data itself is a probability distribution or an empirical distribution), such as the support measure machines (Muandet et al., 2012) and distribution regression methods (Szabó et al., 2015; Jitkrittum et al., 2015). These methods have found applications in a variety of fields, such as astronomy (Muandet and Schölkopf, 2013), ecological inference (Flaxman et al., 2015) and natural language

processing (Yoshikawa et al., 2014).

## 2.3 Characteristic kernels and metrics on distributions

Conditions for kernels to be characteristic have been extensively studied in the literature (Fukumizu et al., 2008, 2009b; Sriperumbudur et al., 2010; Gretton et al., 2012). Here we review these conditions, following Sriperumbudur et al. (2010).

**Shift-invariant kernels on $\mathbb{R}^d$.** Let $k$ be a shift-invariant kernel on $\mathbb{R}^d$, that is, there is a positive definite function $\psi : \mathbb{R}^d \to \mathbb{R}$ such that $k(x, y) = \psi(x - y)$. In this case, a necessary and sufficient condition for the kernel being characteristic is known. Namely, the shift invariant kernel $k$ is characteristic, if and only if the support of the Fourier transform of the function $\psi$ is entire $\mathbb{R}^d$ (Sriperumbudur et al., 2010, Theorem 9).

**Integrally strictly positive definite kernels on general topological spaces.** Let $\mathcal{X}$ be a topological space. A measurable and bounded kernel on $\mathcal{X}$ is defined to be *integrally strictly positive definite*, if for all finite non-zero signed Borel measure $\mu$ on $\mathcal{X}$, we have

$$\int \int_{\mathcal{X}} k(x, y) d\mu x d\mu(y) > 0.$$

It is known that an integrally strictly positive definite kernel is characteristic (Sriperumbudur et al., 2010, Theorem 7).

**Metric on distributions.** We can define a metric on distributions using a characteristic kernel. That is, we can define the distance between any distributions $P$ and $Q$ as the RKHS distance between their kernel means:

$$d_k(P, Q) := \|m_P - m_Q\|_{\mathcal{H}}, \tag{2.4}$$

$\|\cdot\|_{\mathcal{H}}$ is the norm of the RKHS: $\|f\|_{\mathcal{H}} := \sqrt{\langle f, f \rangle_{\mathcal{H}}}$ for all $f \in \mathcal{H}$. It can be easily shown that this satisfies the conditions of a metric: (i) $d_k(P, P) = 0$ for any distribution $P$; (ii) $d_k(P, Q) \leq d_k(P, R) + d_k(R, Q)$ for any distributions $P$, $Q$ and $R$; and (iii) $d_k(P, Q) = 0$ implies $P = Q$. The first and second conditions are consequences of the use of the distance in a Hilbert space (i.e. the RKHS $\mathcal{H}$). The third condition is due to the characteristic property of the kernel. Relations to other metrics on distributions have been also studied (Sriperumbudur et al., 2010).

The distance (2.4) is also called the maximum mean discrepancy (MMD) (Gretton et al., 2012). This is because (2.4) can be written as

$$\|m_P - m_Q\|_{\mathcal{H}} = \sup_{f \in \mathcal{H}: \|f\|_{\mathcal{H}} \leq 1} \int f(x) dP(x) - \int f(x) dQ(x). \tag{2.5}$$

Namely, by computing the kernel distance, we implicitly consider a witness function in the unit ball of the RKHS such that the difference between function value expectations with respect to the two distributions is maximized.

For the kernel distance, there is another characterization in terms of characteristic functions, if the kernel is shift-invariant on $\mathbb{R}^d$. Let $\psi : \mathbb{R}^d \to \mathbb{R}$ be a bounded, continuous positive definite function, such that $k(x, y) = \psi(x - y)$. Then by Bochner's theorem, there is a finite nonnegative measure $\Lambda$ on $\mathbb{R}^d$, such that $\psi$ is given as the Fourier transform of $\Lambda$:

$$\psi(x) = \int e^{-\sqrt{-1} x^T w} d\Lambda(w), \quad x \in \mathbb{R}^d. \tag{2.6}$$

Then we have

$$\|m_P - m_Q\|_{\mathcal{H}} = \sqrt{\int |\phi_P(w) - \phi_Q(w)|^2 d\Lambda(w)}, \tag{2.7}$$

where $\phi_P$ and $\phi_Q$ denote the characteristic functions of $P$ and $Q$, respectively (Sriperumbudur et al., 2010, Corollary 4). In other words, the kernel distance can be written as a weighted $L_2$ distance between the characteristic functions, where the weight function is given by the (inverse) Fourier transform the function $\psi$ that induces the kernel. This is very similar to the so-called the energy distance, which uses the weight function defined as $\frac{1}{\|w\|^{d+1}}$ instead of $\Lambda$ (Székely and Rizzo, 2013, Proposition 1).

## 2.4 Estimation of kernel means

In practice, we want to estimate kernel means from samples. Suppose we are given an i.i.d. sample $X_1, \ldots, X_n$ from a distribution $P$. Define an estimator of $m_P$ by the empirical mean:

$$\hat{m}_P := \frac{1}{n} \sum_{i=1}^{n} k(\cdot, X_i). \tag{2.8}$$

If the kernel $k$ is bounded, then this converges to $m_P$ at a rate $\|\hat{m}_P - m_P\|_{\mathcal{H}} = O_p(n^{-1/2})$ (Smola et al., 2007), where $O_p$ denotes the asymptotic order in probability. Note that this rate is independent of the dimensionality of the space $\mathcal{X}$.

This estimate (2.8) can be used to estimate the the kernel distance (2.4) on distributions. Suppose we have another sample $Y_1, \ldots, Y_n$ from a distribution $Q$. Then we can also define an estimate of the kernel mean $m_Q$ by $\hat{m}_Q := \frac{1}{n}\sum_{i=1}^n k(\cdot, Y_i)$, which converges at a rate $O_p(n^{-1/2})$. Then the kernel distance (2.4) can be estimated by plugging these kernel mean estimates into (2.4). Thanks to the reproducing property of the kernel, this plugin estimate can be analytically computed by just evaluating the kernel values between the samples:

$$\|\hat{m}_P - \hat{m}_Q\|_{\mathcal{H}} = \sqrt{\frac{1}{n^2}\sum_{i,j} k(X_i, X_j) - \frac{2}{mn}\sum_{i,j} k(X_i, Y_j) + \frac{1}{m^2}\sum_{i,j} k(Y_i, Y_j)}. \quad (2.9)$$

This converges to the population quantity (2.4) at a rate $O(n^{-1/2})$, since the kernel mean estimates converge at this rate. For the kernel distance, there also exist statistically or computationally more efficient estimators (Gretton et al., 2012).

## 2.5 Kernel mean embeddings of conditional distributions

### 2.5.1 Conditional mean embeddings

We can also define kernel means for conditional distributions (Song et al., 2009; Grünewälder et al., 2012a; Song et al., 2013). To show this, let $\mathcal{X}$ and $\mathcal{Y}$ be measurable spaces, and $(X_1, Y_1), \ldots, (X_n, Y_n)$ be an i.i.d. sample on $\mathcal{X}\times\mathcal{Y}$ with a joint distribution $p(x, y)$[1]. Let $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ be bounded measurable kernels on $\mathcal{X}$ and $\mathcal{Y}$, respectively. Then the kernel mean of the conditional distribution $p(y|x)$ is defined as

$$m_{Y|x} := \int k_{\mathcal{Y}}(\cdot, y)p(y|x)dy. \quad (2.10)$$

Different from the previous ones, the conditional kernel mean depends on the input $x \in \mathcal{X}$. Therefore it may be seen as a function-valued function. In fact, conditional kernel means can be understood from the viewpoint of function-valued regression (Grünewälder et al., 2012a).

As for the kernel distance, the conditional kernel mean (2.10) can be estimated by simple linear algebraic operations using the joint sample $\{(X_i, Y_i)\}$ Let $G_X = (k_{\mathcal{X}}(X_i, X_j)) \in \mathbb{R}^{n\times n}$ be the kernel matrix on the sample $X_1, \ldots, X_n$. Then we can

[1]For simplicity of notation, we use the density form to express the joint and conditional distributions.

estimate conditional kernel mean (2.10) as

$$\hat{m}_{Y|x} := \sum_{i=1}^{n} w_i k_{\mathcal{Y}}(\cdot, Y_i). \tag{2.11}$$

Here the weights $w_1, \ldots, w_n \in \mathbb{R}$ are computed as

$$(w_1, \ldots, w_n)^T = (G_X + n\lambda I)^{-1} \mathbf{k}_X \in \mathbb{R}^n,$$

where $\lambda > 0$ is a regularization constant and $\mathbf{k}_X := (k_{\mathcal{X}}(x, X_1), \ldots, k_{\mathcal{X}}(x, X_n))^T \in \mathbb{R}^n$. Note that the weights are a function of the input $x \in \mathcal{X}$. To make this estimator consistent, the regularization constant should be decayed to 0 as the sample size increases. It is known that the estimator achieves min-max optimal convergence rates under certain assumptions (Grünewälder et al., 2012a).

### 2.5.2 Kernel Bayes' Rule (KBR)

As an extension of the conditional kernel means, we can consider kernel means for posterior distributions, taking prior distributions into account. Estimators of such posterior kernel means have been developed, known as the Kernel Bayes' Rule (KBR). We next explain these concepts.

Let $p(x, y)$ be a joint probability on the product space $\mathcal{X} \times \mathcal{Y}$ that decomposes as $p(x, y) = p(y|x)p(x)$. Let $\pi(x)$ be a prior distribution on $\mathcal{X}$. Then the conditional probability $p(y|x)$ and the prior $\pi(x)$ define the posterior distribution by Bayes' rule;

$$p^{\pi}(x|y) \propto p(y|x)\pi(x).$$

The assumption here is that the conditional probability $p(y|x)$ is unknown. Instead, we are given an i.i.d. sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ from the joint probability $p(x, y)$. We wish to estimate the posterior $p^{\pi}(x|y)$ using the sample. KBR achieves this by estimating the kernel mean of $p^{\pi}(x|y)$.

Define the kernel means of the prior $\pi(x)$ and the posterior $p^{\pi}(x|y)$:

$$m_{\pi} := \int k_{\mathcal{X}}(\cdot, x)\pi(x)dx, \quad m^{\pi}_{X|y} := \int k_{\mathcal{X}}(\cdot, x)p^{\pi}(x|y)dx.$$

KBR also requires that $m_{\pi}$ be expressed as a weighted sample. Let $\hat{m}_{\pi} := \sum_{j=1}^{\ell} \gamma_j k_{\mathcal{X}}(\cdot, U_j)$ be a sample expression of $m_{\pi}$, where $\ell \in \mathbb{N}$, $\gamma_1, \ldots, \gamma_{\ell} \in \mathbb{R}$ and $U_1, \ldots, U_{\ell} \in \mathcal{X}$. For example, suppose $U_1, \ldots, U_{\ell}$ are i.i.d. drawn from $\pi(x)$. Then $\gamma_j = 1/\ell$ suffices.

Given the joint sample $\{(X_i, Y_i)\}_{i=1}^{n}$ and the empirical prior mean $\hat{m}_{\pi}$, KBR esti-

**Algorithm 1** Kernel Bayes' Rule

1: **Input:** $\mathbf{k}_Y, \mathbf{m}_\pi \in \mathbb{R}^n$, $G_X, G_Y \in \mathbb{R}^{n\times n}$, $\varepsilon, \delta > 0$.
2: **Output:** $w := (w_1, \ldots, w_n)^T \in \mathbb{R}^n$.

3: $\Lambda \leftarrow \mathrm{diag}((G_X + n\varepsilon I_n)^{-1}\mathbf{m}_\pi) \in \mathbb{R}^{n\times n}$.
4: $w \leftarrow \Lambda G_Y((\Lambda G_Y)^2 + \delta I_n)^{-1}\Lambda \mathbf{k}_Y \in \mathbb{R}^n$.

mates the kernel posterior mean $m^\pi_{X|y}$ as a weighted sum of the feature vectors:

$$\hat{m}^\pi_{X|y} := \sum_{i=1}^n w_i k_\mathcal{X}(\cdot, X_i), \tag{2.12}$$

where the weights $w := (w_1, \ldots, w_n)^T \in \mathbb{R}^n$ are given by Algorithm 1. Here $\mathrm{diag}(v)$ for $v \in \mathbb{R}^n$ denotes a diagonal matrix with diagonal entries $v$. It takes as input (i) vectors $\mathbf{k}_Y = (k_\mathcal{Y}(y, Y_1), \ldots, k_\mathcal{Y}(y, Y_n))^T$, $\mathbf{m}_\pi = (\hat{m}_\pi(X_1), \ldots, \hat{m}_\pi(X_n))^T \in \mathbb{R}^n$, where $\hat{m}_\pi(X_i) = \sum_{j=1}^\ell \gamma_j k_\mathcal{X}(X_i, U_j)$; (ii) kernel matrices $G_X = (k_\mathcal{X}(X_i, X_j)), G_Y = (k_\mathcal{Y}(Y_i, Y_j)) \in \mathbb{R}^{n\times n}$; and (iii) regularization constants $\varepsilon, \delta > 0$. The weight vector $w := (w_1, \ldots, w_n)^T \in \mathbb{R}^n$ is obtained by matrix computations involving two regularized matrix inversions. Note that these weights can be negative.

Fukumizu et al. (2013) showed that KBR is a consistent estimator of the kernel posterior mean under certain smoothness assumptions: the estimate (2.12) converges to $m^\pi_{X|y}$, as the sample size goes to infinity $n \to \infty$ and $\hat{m}_\pi$ converges to $m_\pi$ (with $\varepsilon, \delta \to 0$ in appropriate speed). For details, see Fukumizu et al. (2013); Song et al. (2013).

## 2.6 Properties of weight sample estimators

In general, as shown above, a kernel mean $m_P$ is estimated as a weighted sum of feature vectors;

$$\hat{m}_P = \sum_{i=1}^n w_i k(\cdot, X_i), \tag{2.13}$$

with samples $X_1, \ldots, X_n \in \mathcal{X}$ and (possibly negative) weights $w_1, \ldots, w_n \in \mathbb{R}$. Suppose $\hat{m}_P$ is close to $m_P$, i.e., $\|\hat{m}_P - m_P\|_\mathcal{H}$ is small. Then $\hat{m}_P$ is supposed to have accurate information about $P$, as $m_P$ preserves all the information of $P$.

How can we decode the information of $P$ from $\hat{m}_P$? The empirical kernel mean (2.13) has the following property, which is due to the reproducing property of the

kernel:

$$\langle \hat{m}_P, f \rangle_{\mathcal{H}} = \sum_{i=1}^{n} w_i f(X_i), \quad \forall f \in \mathcal{H}. \tag{2.14}$$

Namely, the weighted average of any function in the RKHS is equal to the inner product between the empirical kernel mean and that function. This is analogous to the property (2.3) of the population kernel mean $m_P$. Let $f$ be any function in $\mathcal{H}$. From these properties (2.3) (2.14), we have

$$\left| \mathbf{E}_{X \sim P}[f(X)] - \sum_{i=1}^{n} w_i f(X_i) \right| = |\langle m_P - \hat{m}_P, f \rangle_{\mathcal{H}}| \leq \|m_P - \hat{m}_P\|_{\mathcal{H}} \|f\|_{\mathcal{H}}, \tag{2.15}$$

where we used the Cauchy-Schwartz inequality. Therefore the left hand side will be close to 0, if the error $\|m_P - \hat{m}_P\|_{\mathcal{H}}$ is small. This shows that the expectation of $f$ can be estimated by the weighted average $\sum_{i=1}^{n} w_i f(X_i)$. Note that here $f$ is a function in the RKHS, but the same can also be shown for functions outside the RKHS under certain assumptions when the kernel is Gaussian; this is what we will show in Chapter 5. In this way, the estimator of the form (2.13) provides estimators of moments, probability masses on sets and the density function (if this exists).

## 2.7 Kernel Herding

Finally, we explain the Kernel Herding algorithm (Chen et al., 2010). Different from estimators discussed above, this algorithm assumes that a kernel mean $m_P$ is given. It aims at approximating the kernel mean by a finite sample of possibly small size. In other words, the aim is to generate samples $x_1, x_2, \ldots, x_\ell \in \mathcal{X}$ such that the empirical mean $\check{m}_P := \frac{1}{\ell} \sum_{i=1}^{\ell} k(\cdot, x_i)$ is close to the kernel mean $m_P$ in the RKHS, i.e., the error $\|m_P - \check{m}_P\|_{\mathcal{H}}$ is small.

The samples generated in this way are useful for numerical integration: for any function $f$ in the RKHS, the empirical average $\frac{1}{\ell} \sum_{i=1}^{\ell} f(x_i)$ gives an approximation of the integral $\int f(x) dP(x)$, with an error bounded by $\|f\|_{\mathcal{H}} \|m_P - \hat{m}_P\|_{\mathcal{H}}$. This follows from (2.15). Approaches to generate such samples include Quasi Monte Carlo methods; see (Dick et al., 2013).

Kernel Herding is one approach for this purpose. It generates samples $x_1, \ldots, x_\ell$ deterministically and greedily, by solving the following optimization problems:

$$x_1 = \arg\max_{x \in \mathcal{X}} \; m_P(x), \tag{2.16}$$

$$x_\ell = \arg\max_{x \in \mathcal{X}} \; m_P(x) - \frac{1}{\ell} \sum_{i=1}^{\ell-1} k(x, x_i), \quad (\ell \geq 2) \tag{2.17}$$

where $m_P(x)$ denotes the evaluation of $m_P$ at $x$ (recall that $m_P$ is a function in $\mathcal{H}$).

An intuitive interpretation of this procedure can be given if there is a constant $R > 0$ such that $k(x, x) = R$ for all $x \in \mathcal{X}$ (e.g., $R = 1$ if $k$ is Gaussian). Suppose that $x_1, \ldots, x_{\ell-1}$ are already calculated. In this case, it can be shown that $x_\ell$ in (2.17) is the minimizer of

$$\mathcal{E}_\ell \;:=\; \left\| m_P - \frac{1}{\ell} \sum_{i=1}^{\ell} k(\cdot, x_i) \right\|_{\mathcal{H}}. \tag{2.18}$$

Thus, Kernel Herding performs greedy minimization of the distance between $m_P$ and the empirical kernel mean $\breve{m}_P = \frac{1}{\ell} \sum_{i=1}^{\ell} k(\cdot, x_i)$.

It can be shown that the error $\mathcal{E}_\ell$ of (2.18) decreases at a rate at least $O(\ell^{-1/2})$ under the assumption that $k$ is bounded (Bach et al., 2012). In other words, the herding samples $x_1, \ldots, x_\ell$ provide a convergent approximation of $m_P$. In this sense, Kernel Herding can be seen as a (pseudo) sampling method. Note that $m_P$ itself can be an empirical kernel mean of the form (2.13). These properties are important for our resampling algorithm developed in Section 3.2.

It should be noted that $\mathcal{E}_\ell$ decreases at a faster rate $O(\ell^{-1})$ under a certain assumption (Chen et al., 2010): this is much faster than the rate of $\ell$ i.i.d. samples $O(\ell^{-1/2})$. Unfortunately, this assumption only holds when $\mathcal{H}$ is finite dimensional (Bach et al., 2012), and therefore the fast rate of $O(\ell^{-1})$ has not been guaranteed for infinite dimensional cases.

# Chapter 3

# Sampling and resampling with kernel mean embeddings

In this chapter, we discuss the use of sampling with empirical kernel means. As we saw in Chapter 2, in general an empirical kernel mean is given as a weighted sum of feature vectors. This expression is similar to that of an empirical distribution, which is given as a weighted sum of delta functions. In Monte Carlo methods, this representation is combined with sampling methods to realize inference in graphical models. One of the most successful applications is particle filters, where sampling is employed to compute forward probabilities. In this chapter we investigate the applicability of this sampling procedure to an empirical kernel mean. Algorithms presented in this chapter serve as building blocks of the filtering method in Chapter 4.

In Section 3.1, we formulate this sampling procedure in terms of empirical kernel means. We present a theoretical justification, and discuss factors that affect the estimation accuracy of sampling. This reveals that the quantity called *effective sample size* plays an important role. In Section 3.2, we present a resampling algorithm based on Kernel Herding. This algorithm is motivated by the analysis in Section 3.1, and is proposed for the purpose of improving the accuracy of the sampling procedure. In Section 3.3, we explain in more detail the mechanism of resampling. In Section 3.4, we theoretically analyze the proposed resampling algorithm. This analysis presents a novel convergence result of Kernel Herding, which may be of independent interest. In Section 3.5, we conduct toy experiments to empirically confirm the theoretical results. All proofs are presented in Section 3.6.

## 3.1 Sampling algorithm

We begin by introducing the notation. Let $\mathcal{X}$ be a measurable space, and $P$ be a probability distribution on $\mathcal{X}$. Let $p(\cdot|x)$ be a conditional distribution on $\mathcal{X}$ conditioned on $x \in \mathcal{X}$. Let $Q$ be a marginal distribution on $\mathcal{X}$ defined by $Q(B) = \int p(B|x)dP(x)$ for all measurable $B \subset \mathcal{X}$.[1]

Let $k_{\mathcal{X}}$ be a positive definite kernel on $\mathcal{X}$, and $\mathcal{H}_{\mathcal{X}}$ be the RKHS associated with $k_{\mathcal{X}}$. Let $m_P = \int k_{\mathcal{X}}(\cdot, x)dP(x)$ and $m_Q = \int k_{\mathcal{X}}(\cdot, x)dQ(x)$ be the kernel means of $P$ and $Q$, respectively. Suppose that we are given an empirical estimate of $m_P$ as

$$\hat{m}_P := \sum_{i=1}^{n} w_i k_{\mathcal{X}}(\cdot, X_i), \tag{3.1}$$

where $w_1, \ldots, w_n \in \mathbb{R}$ and $X_1, \ldots, X_n \in \mathcal{X}$. Based on this, we wish to estimate the kernel mean $m_Q$.

We consider the following sampling procedure with the conditional distribution: for each sample $X_i$, we generate a new sample $X'_i$ with the conditional distribution $X'_i \sim p(\cdot|X_i)$. Then we estimate $m_Q$ by

$$\hat{m}_Q := \sum_{i=1}^{n} w_i k_{\mathcal{X}}(\cdot, X'_i). \tag{3.2}$$

The following theorem provides an upper-bound on the error of (3.2), and reveals properties of (3.1) that affect the error of the estimator (3.2). The proof is given in Section 3.6.1.

**Theorem 1.** *Let $\hat{m}_P$ be a fixed estimate of $m_P$ given by (3.1). Define a function $\theta$ on $\mathcal{X} \times \mathcal{X}$ by $\theta(x_1, x_2) = \int\int k_{\mathcal{X}}(x'_1, x'_2)dp(x'_1|x_1)dp(x'_2|x_2), \forall x_1, x_2 \in \mathcal{X} \times \mathcal{X}$, and assume that $\theta$ is included in the tensor RKHS $\mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}$.[2] The estimator $\hat{m}_Q$ (3.2)*

[1] We can consider another measurable space $\mathcal{Y}$ for the output variable. Namely, $p(\cdot|x)$ can be a conditional distribution on $\mathcal{Y}$ conditioned on $x \in \mathcal{X}$, and $Q$ can be the resulting marginal distribution on $\mathcal{Y}$. Here, however, we restrict ourselves to the setting $\mathcal{X} = \mathcal{Y}$ for simplicity of notation. Note also that this setting is sufficient for the application to state-space models in Chapter 4.

[2] The tensor RKHS $\mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}$ is the RKHS of a product kernel $k_{\mathcal{X}\times\mathcal{X}}$ on $\mathcal{X} \times \mathcal{X}$ defined as $k_{\mathcal{X}\times\mathcal{X}}((x_a, x_b), (x_c, x_d)) = k_{\mathcal{X}}(x_a, x_c)k_{\mathcal{X}}(x_b, x_d), \forall (x_a, x_b), (x_c, x_d) \in \mathcal{X} \times \mathcal{X}$. This space $\mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}$ consists of smooth functions on $\mathcal{X} \times \mathcal{X}$, if the kernel $k_{\mathcal{X}}$ is smooth (e.g., if $k_{\mathcal{X}}$ is Gaussian; see Sec. 4 of Steinwart and Christmann (2008)). In this case, we can interpret this assumption as requiring that $\theta$ be smooth as a function on $\mathcal{X} \times \mathcal{X}$.

The function $\theta$ can be written as the inner product between the kernel means of the conditional distributions: $\theta(x_1, x_2) = \langle m_{p(\cdot|x_1)}, m_{p(\cdot|x_2)} \rangle_{\mathcal{H}_{\mathcal{X}}}$, where $m_{p(\cdot|x)} := \int k_{\mathcal{X}}(\cdot, x')dp(x'|x)$. Therefore the assumption may be further seen as requiring that the map $x \to m_{p(\cdot|x)}$ be smooth. Note that while similar assumptions are common in the literature on kernel mean embeddings (e.g., Theorem 5 of

*then satisfies*

$$\begin{aligned}
&\mathbf{E}_{X'_1,\ldots,X'_n}[\|\hat{m}_Q - m_Q\|^2_{\mathcal{H}_\mathcal{X}}] \\
&\leq \sum_{i=1}^n w_i^2 (\mathbf{E}_{X'_i}[k_\mathcal{X}(X'_i, X'_i)] - \mathbf{E}_{X'_i,\tilde{X}'_i}[k_\mathcal{X}(X'_i, \tilde{X}'_i)]) \qquad (3.3) \\
&\quad + \|\hat{m}_P - m_P\|^2_{\mathcal{H}_\mathcal{X}} \|\theta\|_{\mathcal{H}_\mathcal{X}\otimes\mathcal{H}_\mathcal{X}}, \qquad (3.4)
\end{aligned}$$

*where $X'_i \sim p(\cdot|X_i)$ and $\tilde{X}'_i$ is an independent copy of $X'_i$.*

From Theorem 1, we can make the following observations. First, the second term (3.4) of the upper-bound shows that the error of the estimator (3.2) is likely to be large if the given estimate (3.1) has large error $\|\hat{m}_P - m_P\|^2_{\mathcal{H}_\mathcal{X}}$, which is reasonable to expect.

Second, the first term (3.3) shows that the error of (3.2) can be large if the distribution of $X'_i$ (i.e. $p(\cdot|X_i)$) has large variance. For example, suppose $X'_i = f(X_i) + \varepsilon_i$, where $f : \mathcal{X} \to \mathcal{X}$ is some mapping and $\varepsilon_i$ is a random variable with mean 0. Let $k_\mathcal{X}$ be the Gaussian kernel: $k_\mathcal{X}(x, x') = \exp(-\|x - x'\|/2\alpha)$ for some $\alpha > 0$. Then $\mathbf{E}_{X'_i}[k_\mathcal{X}(X'_i, X'_i)] - \mathbf{E}_{X'_i,\tilde{X}'_i}[k_\mathcal{X}(X'_i, \tilde{X}'_i)]$ increases from 0 to 1, as the variance of $\varepsilon_i$ (i.e. the variance of $X'_i$) increases from 0 to infinity. Therefore in this case (3.3) is upper-bounded at worst by $\sum_{i=1}^n w_i^2$. Note that $\mathbf{E}_{X'_i}[k_\mathcal{X}(X'_i, X'_i)] - \mathbf{E}_{X'_i,\tilde{X}'_i}[k_\mathcal{X}(X'_i, \tilde{X}'_i)]$ is always non-negative.[3]

**Weight variance and effective sample size.** Now let us assume that the kernel $k_\mathcal{X}$ is bounded, i.e., there is a constant $C > 0$ such that $\sup_{x\in\mathcal{X}} k_\mathcal{X}(x, x) < C$. Then the inequality of Theorem 1 can be further bounded as

$$\mathbf{E}_{X'_1,\ldots,X'_n}[\|\hat{m}_Q - m_Q\|^2_{\mathcal{H}_\mathcal{X}}] \leq 2C \sum_{i=1}^n w_i^2 + \|\hat{m}_P - m_P\|^2_{\mathcal{H}_\mathcal{X}} \|\theta\|_{\mathcal{H}_\mathcal{X}\otimes\mathcal{H}_\mathcal{X}}. \qquad (3.5)$$

This bound shows that two quantities are important in the estimate (3.1): (i) the sum of squared weights $\sum_{i=1}^n w_i^2$, and (ii) the error $\|\hat{m}_P - m_P\|^2_{\mathcal{H}_\mathcal{X}}$. In other words, the error of (3.2) can be large if the quantity $\sum_{i=1}^n w_i^2$ is large, regardless of the accuracy

Fukumizu et al. (2013)), we may relax this assumption by using approximate arguments in learning theory (e.g., Theorem 2.2 and 2.3 of Eberts and Steinwart (2013)). This analysis remains a topic for future research.

[3]To show this, it is sufficient to prove that $\int\int k_\mathcal{X}(x, \tilde{x}) dP(x) dP(\tilde{x}) \leq \int k_\mathcal{X}(x, x) dP(x)$ for any probability $P$. This can be shown as follows. $\int\int k_\mathcal{X}(x, \tilde{x}) dP(x) dP(\tilde{x}) = \int\int \langle k_\mathcal{X}(\cdot, x), k_\mathcal{X}(\cdot, \tilde{x})\rangle_{\mathcal{H}_\mathcal{X}} dP(x) dP(\tilde{x}) \leq \int\int \sqrt{k_\mathcal{X}(x, x)}\sqrt{k_\mathcal{X}(\tilde{x}, \tilde{x})} dP(x) dP(\tilde{x}) \leq \int k_\mathcal{X}(x, x) dP(x)$. Here we used the reproducing property, the Cauchy-Schwartz inequality and Jensen's inequality.

**Algorithm 2** Resampling with Kernel Herding

1: **Input:** $\{(w_i, X_i)\}_{i=1}^n$.
2: **Output:** $\bar{X}_1, \ldots, \bar{X}_n \in \{X_i\}_{i=1}^n$.
3: **Requirement:** $k_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

4: $\bar{X}_1 \leftarrow \underset{x \in \{X_1, \ldots, X_n\}}{\arg\max} \sum_{i=1}^n w_i k_{\mathcal{X}}(x, X_i)$.
5: **for** $p = 2$ to $n$ **do**
6: $\quad \bar{X}_p \leftarrow \underset{x \in \{X_1, \ldots, X_n\}}{\arg\max} \sum_{i=1}^n w_i k_{\mathcal{X}}(x, X_i) - \frac{1}{p} \sum_{j=1}^{p-1} k_{\mathcal{X}}(x, \bar{X}_j)$
7: **end for**

of (3.1) as an estimator of $m_P$. In fact, the estimator of the form (3.1) can have large $\sum_{i=1}^n w_i^2$ even when $\|\hat{m}_P - m_P\|_{\mathcal{H}_{\mathcal{X}}}^2$ is small, as shown in Section 3.5.

The quantity $\sum_{i=1}^n w_i^2$ essentially represents the *variance* of the weights $w_1, \ldots, w_n$. Therefore it takes a large value when the weight variance is large. This happens, for example, when the mass of the weights concentrates on a few samples, and the rest of them are close to 0. Figure 3.1 (left) describes such a situation.

In particle methods, this quantity $\sum_{i=1}^n w_i^2$ also plays an important role under the name of *Effective Sample Size (ESS)* (see, e.g., Sec. 2.5.3 of Liu (2001) and Sec. 3.5 of Doucet and Johansen (2011)). ESS is defined as $1/\sum_{i=1}^n w_i^2$, and represents an actual number of samples that contribute the estimation of a probability. For example, suppose that the weights are normalized, i.e., $\sum_{i=1}^n w_i = 1$. Then ESS is $n$ when the weights are uniform, while it is small when the mass of the weights concentrate on a few samples. Therefore the bound (3.5) can be interpreted as follows: to make (3.2) a good estimator of $m_Q$, we need to have (3.1) such that the ESS is large and the error $\|\hat{m}_P - m_P\|_{\mathcal{H}}$ is small.

## 3.2 Resampling algorithm

Here we introduce a resampling algorithm to improve the accuracy of the sampling procedure. We discuss how it works in Section 3.3. The arguments in the previous section suggests that the estimation accuracy of the sampling procedure can be improved by increasing the effective sample size. Thus our resampling algorithm aims to increase the effective sample size. The algorithm is based on Kernel Herding in Section 2.7.

The procedure is summarized in Algorithm 2. Specifically, we generate each $\bar{X}_i$ by searching the solution of the optimization problem in (2.16) (2.17) from a finite set of samples $\{X_1, \ldots, X_n\}$ in (3.1). We allow repetitions in $\bar{X}_1, \ldots, \bar{X}_n$. We can expect that the resulting empirical kernel mean $\check{m}_P := \frac{1}{n} \sum_{i=1}^n k_{\mathcal{X}}(\cdot, \bar{X}_i)$ is close to

$m_P$ in the RKHS if the samples $X_1, \ldots, X_n$ cover the support of $P$ sufficiently. This is verified by the theoretical analysis of Section 3.4.

Here searching for the solutions from a finite set reduces the computational costs of Kernel Herding. It is possible to search from the entire space $\mathcal{X}$, if we have sufficient time or if the sample size $n$ is small enough; it depends on applications and available computational resources. We also note that the size of the resampling samples is not necessarily $n$; this depends on how accurately these samples approximate (3.1). Thus a smaller number of samples may be sufficient. In this case we can reduce the computational costs of resampling, as discussed in Section 3.3.

The aim of our resampling step is similar to that of the resampling step of a particle filter (see, e.g., Doucet and Johansen (2011)). Intuitively, the aim is to eliminate samples with very small weights, and replicate those with large weights (see Figure 3.1). In particle methods, this is realized by generating samples from the empirical distribution defined by a weighted sample (therefore this procedure is called "resampling"). Our resampling step is a realization of such a procedure in terms of the kernel mean embedding: we generate samples $\bar{X}_1, \ldots, \bar{X}_n$ from the empirical kernel mean (3.1).

Note that the resampling algorithm of particle methods is not appropriate for use with kernel mean embeddings. This is because it assumes that weights are positive, but our weights in (3.1) can be negative, as this is a kernel mean estimator. One may apply the resampling algorithm of particle methods by first truncating the samples with negative weights. However, there is no guarantee that samples obtained by this heuristic produce a good approximation of (3.1) as a kernel mean, as shown by experiments in Section 3.5. In this sense, the use of Kernel Herding is more natural since it generates samples that approximate a kernel mean.

## 3.3 Role of resampling

In this section, we discuss how the proposed resampling algorithm improves the estimation accuracy of the sampling procedure. By applying Algorithm 2 to the empirical kernel mean $\hat{m}_P$, we obtain new samples $\bar{X}_1, \ldots, \bar{X}_n$. These samples then provide a new estimate of $m_P$ with uniform weights;

$$\check{m}_P = \frac{1}{n} \sum_{i=1}^{n} k_{\mathcal{X}}(\cdot, \bar{X}_i). \tag{3.6}$$

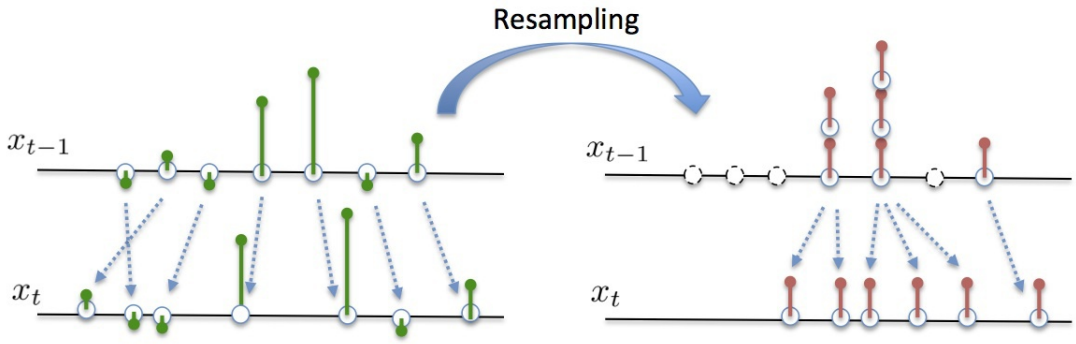



Figure 3.1: An illustration of the sampling procedure with (right) and without (left) the resampling algorithm. The left figure corresponds to the kernel mean estimators (3.1) (3.2) in Section 3.1, and the right one corresponds to those (3.6) (3.7) in Section 3.3

We apply the sampling procedure to this empirical kernel mean: we independently generate a sample $\bar{X}'_i \sim p(\cdot|\bar{X}_i)$ for each $\bar{X}_i$ $(i = 1, \ldots, n)$, and estimate $m_Q$ as

$$\check{m}_Q = \frac{1}{n}\sum_{i=1}^{n} k_{\mathcal{X}}(\cdot, \bar{X}'_i). \tag{3.7}$$

Theorem 1 gives the following bound for this estimator that corresponds to (3.5):

$$\mathbf{E}_{\bar{X}'_1,\ldots,\bar{X}'_n}[\|\check{m}_Q - m_Q\|^2_{\mathcal{H}_{\mathcal{X}}}] \leq \frac{2C}{n} + \|\check{m}_P - m_P\|^2_{\mathcal{H}}\|\theta\|_{\mathcal{H}_{\mathcal{X}}\otimes\mathcal{H}_{\mathcal{X}}}. \tag{3.8}$$

A comparison of the upper-bounds of (3.5) and (3.8) implies that the resampling step is beneficial when (i) $\sum_{i=1}^n w_i^2$ is large (i.e., the ESS is small), and (ii) $\|\check{m}_P - \hat{m}_P\|_{\mathcal{H}_{\mathcal{X}}}$ is small. The condition on $\|\check{m}_P - \hat{m}_P\|_{\mathcal{H}_{\mathcal{X}}}$ means that the loss by Kernel Herding (in terms of the RKHS distance) is small. This implies $\|\hat{m}_P - m_P\|_{\mathcal{H}_{\mathcal{X}}} \approx \|\check{m}_P - m_P\|_{\mathcal{H}_{\mathcal{X}}}$, so the second term of (3.8) is close to that of (3.5). On the other hand, the first term of (3.8) will be much smaller than that of (3.5), if $\sum_{i=1}^n w_i^2 \gg 1/n$. In other words, the resampling algorithm improves the sampling procedure in accuracy by reducing the variance of the weights (i.e., by increasing the ESS). This is illustrated in Figure 3.1.

The above observations lead to the following procedures:

**When to apply resampling.** If $\sum_{i=1}^n w_i^2$ is not large, the gain by the resampling step will be small. Therefore the resampling algorithm should be applied when

**Algorithm 3** Generalized version of Algorithm 2

**Input:** $\hat{m}_P \in \mathcal{H}_\mathcal{X}$, $\{Z_1, \dots, Z_N\} \subset \mathcal{X}$, $\ell \in \mathbb{N}$.
**Output:** $\bar{X}_1, \dots, \bar{X}_\ell \in \{Z_1, \dots, Z_N\}$.

$\bar{X}_1 \leftarrow \underset{x \in \{Z_1, \dots, Z_N\}}{\arg\max}\ \hat{m}_P(x)$.
**for** $p = 2$ to $\ell$ **do**
  $\bar{X}_p \leftarrow \underset{x \in \{Z_1, \dots, Z_N\}}{\arg\max}\ \hat{m}_P(x) - \frac{1}{p}\sum_{j=1}^{p-1} k_\mathcal{X}(x, \bar{X}_j)$
**end for**

$\sum_{i=1}^n w_i^2$ is above a certain threshold, say $2/n$. The same strategy has been commonly used in particle methods (see, e.g., Doucet and Johansen (2011)).

Also, the bound (3.3) of Theorem 1 shows that resampling is not beneficial if the variance of the conditional distribution $p(\cdot|x)$ is very small (i.e., if the conditional distribution is nearly deterministic). In this case, the error of the sampling procedure may increase due to the loss $\|\check{m}_P - \hat{m}_P\|_{\mathcal{H}_\mathcal{X}}$ caused by Kernel Herding.

**Reduction of computational cost.** Algorithm 2 generates $n$ samples $\bar{X}_1, \dots, \bar{X}_n$ with time complexity $O(n^3)$. Suppose that the first $\ell$ samples $\bar{X}_1, \dots, \bar{X}_\ell$, where $\ell < n$, already approximate $\hat{m}_P$ well: $\|\frac{1}{\ell}\sum_{i=1}^\ell k_\mathcal{X}(\cdot, \bar{X}_i) - \hat{m}_P\|_{\mathcal{H}_\mathcal{X}}$ is small. We do not then need to generate the rest of samples $\bar{X}_{\ell+1}, \dots, \bar{X}_n$: we can make $n$ samples by copying the $\ell$ samples $n/\ell$ times (suppose $n$ can be divided by $\ell$ for simplicity, say $n = 2\ell$). Let $\bar{X}_1, \dots, \bar{X}_n$ denote these $n$ samples. Then $\frac{1}{\ell}\sum_{i=1}^\ell k_\mathcal{X}(\cdot, \bar{X}_i) = \frac{1}{n}\sum_{i=1}^n k_\mathcal{X}(\cdot, \bar{X}_i)$ by definition, so $\|\frac{1}{n}\sum_{i=1}^n k_\mathcal{X}(\cdot, \bar{X}_i) - \hat{m}_P\|_{\mathcal{H}_\mathcal{X}}$ is also small. This reduces the time complexity of Algorithm 2 to $O(n^2\ell)$.

One might think that it is unnecessary to copy $n/\ell$ times to make $n$ samples. This is not true, however. Suppose that we just use the first $\ell$ samples to define $\check{m}_P = \frac{1}{\ell}\sum_{i=1}^\ell k_\mathcal{X}(\cdot, \bar{X}_i)$. Then the first term of (3.8) becomes $2C/\ell$, which is larger than $2C/n$ of $n$ samples. This difference involves sampling with the conditional distribution: $\bar{X}_i' \sim p(\cdot|\bar{X}_i)$. If we just use the $\ell$ samples, sampling is done $\ell$ times. If we use the copied $n$ samples, sampling is done $n$ times. Thus the benefit of making $n$ samples comes from sampling with the conditional distribution many times. This matches the bound of Theorem 1, where the first term involves the variance of the conditional distribution.

## 3.4 Convergence rates for resampling

Our resampling algorithm (Algorithm 2) is an approximate version of Kernel Herding in Section 2.7: Algorithm 2 searches for the solutions of the update equations

(2.16) (2.17) from a finite set $\{X_1, \ldots, X_n\} \subset \mathcal{X}$, not from the entire space $\mathcal{X}$. Therefore existing theoretical guarantees for Kernel Herding (Chen et al., 2010; Bach et al., 2012) do not apply to Algorithm 2. Here we provide a theoretical justification.

**Generalized version.** We consider a slightly generalized version shown in Algorithm 3: It takes as input (i) a kernel mean estimator $\hat{m}_P$ of a kernel mean $m_P$, (ii) candidate samples $Z_1, \ldots, Z_N$, and (iii) the number $\ell$ of resampling; It then outputs resampling samples $\bar{X}_1, \ldots, \bar{X}_\ell \in \{Z_1, \ldots, Z_N\}$, which form a new estimator $\check{m}_P := \frac{1}{\ell} \sum_{i=1}^{\ell} k_{\mathcal{X}}(\cdot, \bar{X}_i)$. Here $N$ is the number of the candidate samples.

Algorithm 3 searches for the solutions of the update equations (2.16) (2.17) from the candidate set $\{Z_1, \ldots, Z_N\}$. Note that here these samples $Z_1, \ldots, Z_N$ can be different from those expressing the estimator $\hat{m}_P$. If they are the same, i.e., if the estimator is expressed as $\hat{m}_P = \sum_{i=1}^{n} w_{t,i} k(\cdot, X_i)$ with $n = N$ and $X_i = Z_i$ $(i = 1, \ldots, n)$, then Algorithm 3 reduces to Algorithm 2. In fact, Theorem 2 below allows $\hat{m}_P$ to be any element in the RKHS.

**Convergence rates in terms of $N$ and $\ell$.** Algorithm 3 gives the new estimator $\check{m}_P$ of the kernel mean $m_P$. The error of this new estimator $\|\check{m}_P - m_P\|_{\mathcal{H}_\mathcal{X}}$ should be close to that of the given estimator, $\|\hat{m}_P - m_P\|_{\mathcal{H}_\mathcal{X}}$. Theorem 2 below guarantees this. In particular, it provides convergence rates of $\|\check{m}_P - m_P\|_{\mathcal{H}_\mathcal{X}}$ approaching $\|\hat{m}_P - m_P\|_{\mathcal{H}_\mathcal{X}}$, as $N$ and $\ell$ go to infinity. This theorem follows from Theorem 3 in Section 3.6.2, which holds under weaker assumptions.

**Theorem 2.** *Let $m_P$ be the kernel mean of a distribution $P$, and $\hat{m}_P$ be any element in the RKHS $\mathcal{H}_\mathcal{X}$. Let $Z_1, \ldots, Z_N$ be an i.i.d. sample from a distribution with density $q$. Assume that $P$ has a density function $p$ such that $\sup_{x \in \mathcal{X}} p(x)/q(x) < \infty$. Let $\bar{X}_1, \ldots, \bar{X}_\ell$ be samples given by Algorithm 3 applied to $\hat{m}_P$ with candidate samples $\{Z_1, \ldots, Z_N\}$. Then for $\check{m}_P := \frac{1}{\ell} \sum_{i=1}^{\ell} k(\cdot, \bar{X}_i)$ we have*

$$\|\check{m}_P - m_P\|_{\mathcal{H}_\mathcal{X}}^2 = \left(\|\hat{m}_P - m_P\|_{\mathcal{H}_\mathcal{X}} + O_p(N^{-1/2})\right)^2 + O\left(\frac{\ln \ell}{\ell}\right). \quad (N, \ell \to \infty) \quad (3.9)$$

Our proof in Section 3.6.2 relies on the fact that Kernel Herding can be seen as the Frank-Wolfe optimization method (Bach et al., 2012). Indeed, the error $O(\ln \ell / \ell)$ in (3.9) comes from the optimization error of the Frank-Wolfe method after $\ell$ iterations (Freund and Grigas, 2014, Bound 3.2). On the other hand, the error $O_p(N^{-1/2})$ is due to the approximation of the solution space by a finite set $\{Z_1, \ldots, Z_N\}$. These errors will be small if $N$ and $\ell$ are large enough and the error of the given estimator $\|\hat{m}_P - m_P\|_{\mathcal{H}_\mathcal{X}}$ is relatively large. This is formally stated in Corollary 1 below.

Theorem 2 assumes that the candidate samples are i.i.d. with a density $q$. The assumption $\sup_{x \in \mathcal{X}} p(x)/q(x) < \infty$ requires that the support of $q$ contains that of $p$.

This is a formal characterization of the explanation in Section 3.2 that the samples $X_1, \dots, X_N$ should cover the support of $P$ sufficiently. Note that the statement of Theorem 2 also holds for non i.i.d. candidate samples, as shown in Theorem 3 of Section 3.6.2.

**Convergence rates as $\hat{m}_P$ goes to $m_P$.** Theorem 2 provides convergence rates when the estimator $\hat{m}_P$ is fixed. In Corollary 1 below, we let $\hat{m}_P$ approach $m_P$, and provide convergence rates for $\check{m}_P$ of Algorithm 3 approaching $m_P$. This corollary directly follows from Theorem 2, since the constant terms in $O_p(N^{-1/2})$ and $O(\ln \ell/\ell)$ in (3.9) do not depend on $\hat{m}_P$, which can be seen from the proof in Section 3.6.2.

**Corollary 1.** *Assume that $P$ and $Z_1, \dots, Z_N$ satisfy the conditions in Theorem 2 for all $N$. Let $\hat{m}_P^{(n)}$ be an estimator of $m_P$ such that $\|\hat{m}_P^{(n)} - m_P\|_{\mathcal{H}_\mathcal{X}} = O_p(n^{-b})$ as $n \to \infty$ for some constant $b > 0$.*[4] *Let $N = \ell = \lceil n^{2b} \rceil$. Let $\bar{X}_1^{(n)}, \dots, \bar{X}_\ell^{(n)}$ be samples given by Algorithm 3 applied to $\hat{m}_P^{(n)}$ with candidate samples $\{Z_1, \dots, Z_N\}$. Then for $\check{m}_P^{(n)} := \frac{1}{\ell} \sum_{i=1}^{\ell} k_\mathcal{X}(\cdot, \bar{X}_i^{(n)})$, we have*

$$\|\check{m}_P^{(n)} - m_P\|_{\mathcal{H}_\mathcal{X}} = O_p(n^{-b}) \quad (n \to \infty). \tag{3.10}$$

Corollary 1 assumes that the estimator $\hat{m}_P^{(n)}$ converges to $m_P$ at a rate $O_p(n^{-b})$ for some constant $b > 0$. Then the resulting estimator $\check{m}_P^{(n)}$ by Algorithm 3 also converges to $m_P$ at the same rate $O(n^{-b})$, if we set $N = \ell = \lceil n^{2b} \rceil$. This implies that if we use sufficiently large $N$ and $\ell$, the errors $O_p(N^{-1/2})$ and $O(\ln \ell/\ell)$ in (3.9) can be negligible, as stated earlier. Note that $N = \ell = \lceil n^{2b} \rceil$ implies that $N$ and $\ell$ can be smaller than $n$, since typically we have $b \leq 1/2$ ($b = 1/2$ corresponds to the convergence rates of parametric models). This provides a support for the discussion in Section 3.3 (reduction of computational cost).

**Convergence rates of sampling after resampling.** We can derive convergence rates of the estimator $\check{m}_Q$ (3.7) in Section 3.3. Here we consider the following construction of $\check{m}_Q$ as discussed in Section 3.3 (reduction of computational cost): (i) First apply Algorithm 3 to $\hat{m}_P^{(n)}$, and obtain resampling samples $\bar{X}_1^{(n)}, \dots, \bar{X}_\ell^{(n)} \in \{Z_1, \dots, Z_N\}$; (ii) Copy these samples $\lceil n/\ell \rceil$ times, and let $\bar{X}_1^{(n)}, \dots, \bar{X}_{\ell\lceil n/\ell \rceil}^{(n)}$ be the resulting $\ell \times \lceil n/\ell \rceil$ samples; (iii) Sample with the conditional distribution $\bar{X}_i^{\prime(n)} \sim p(\cdot|\bar{X}_i)$ $(i = 1, \dots, \ell\lceil n/\ell \rceil)$, and define

$$\check{m}_Q^{(n)} := \frac{1}{\ell\lceil n/\ell \rceil} \sum_{i=1}^{\ell\lceil n/\ell \rceil} k_\mathcal{X}(\cdot, \bar{X}_i^{\prime(n)}). \tag{3.11}$$

[4] Here the estimator $\hat{m}_P^{(n)}$ and the candidate samples $Z_1, \dots, Z_N$ can be dependent.

The following corollary is a consequence of Corollary 1, Theorem 1 and the bound (3.8). Note that Theorem 1 obtains convergence in expectation, which implies convergence in probability.

**Corollary 2.** *Let $\theta$ be the function defined in Theorem 1 and assume $\theta \in \mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}$. Assume that $P$ and $Z_1, \ldots, Z_N$ satisfy the conditions in Theorem 2 for all $N$. Let $\hat{m}_P^{(n)}$ be an estimator of $m_P$ such that $\|\hat{m}_P^{(n)} - m_P\|_{\mathcal{H}_\mathcal{X}} = O_p(n^{-b})$ as $n \to \infty$ for some constant $b > 0$. Let $N = \ell = \lceil n^{2b} \rceil$. Then for the estimator $\check{m}_Q^{(n)}$ defined as (3.11), we have*

$$\|\check{m}_Q^{(n)} - m_Q\|_{\mathcal{H}_\mathcal{X}} = O_p(n^{-\min(b,1/2)}) \quad (n \to \infty).$$

Suppose $b \leq 1/2$, which holds with basically any nonparametric estimators. Then Corollary 2 shows that the estimator $\hat{m}_Q^{(n)}$ achieves the same convergence rate as the input estimator $\hat{m}_P^{(n)}$. Note that without resampling, the rate becomes $O_p(\sqrt{\sum_{i=1}^n (w_i^{(n)})^2} + n^{-b})$, where the weights are given by the input estimator $\hat{m}_P^{(n)} := \sum_{i=1}^n w_i^{(n)} k_\mathcal{X}(\cdot, X_i^{(n)})$ (see the bound (3.5)). Thanks to resampling, (the square root of) the sum of the squared weights in the case of Corollary 2 becomes $1/\sqrt{\ell \lceil n/\ell \rceil} \leq 1/\sqrt{n}$, which is usually smaller than $\sqrt{\sum_{i=1}^n (w_i^{(n)})^2}$ and is faster than or equal to $O_p(n^{-b})$. This shows the merit of resampling in terms of convergence rates; see also the discussions in Section 3.3.

## 3.5 Experiments

Here we conduct toy experiments to look at how the sampling and resampling procedures work. Specifications of the problem are described below.

We consider the setting $\mathcal{X} = \mathbb{R}$. We will need to evaluate the errors $\|m_P - \hat{m}_P\|_{\mathcal{H}_\mathcal{X}}$ and $\|m_Q - \hat{m}_Q\|_{\mathcal{H}_\mathcal{X}}$, so we need to know the true kernel means $m_P$ and $m_Q$. To this end, we define the distributions and the kernel to be Gaussian: this allows us to obtain analytic expressions for $m_P$ and $m_Q$.

**Distributions and kernel.** More specifically, we define the marginal $P$ and the conditional distribution $p(\cdot|x)$ to be Gaussian: $P = \mathbb{N}(0, \sigma_P^2)$ and $p(\cdot|x) = \mathbb{N}(x, \sigma_{\text{cond}}^2)$. Then the resulting $Q = \int p(\cdot|x) dP(x)$ also becomes Gaussian: $Q = \mathbb{N}(0, \sigma_P^2 + \sigma_{\text{cond}}^2)$. We define $k_\mathcal{X}$ to be the Gaussian kernel: $k_\mathcal{X}(x, x') = \exp(-(x - x')^2/2\gamma^2)$. We set $\sigma_P = \sigma_{\text{cond}} = \gamma = 0.1$.

**Kernel means.** Due to the convolution theorem of Gaussian functions, the kernel means $m_P = \int k_\mathcal{X}(\cdot, x) dP(x)$ and $m_Q = \int k_\mathcal{X}(\cdot, x) dQ(x)$ can be analytically com-

puted: $m_P(x) = \sqrt{\frac{\gamma^2}{\sigma^2+\gamma^2}} \exp(-\frac{x^2}{2(\gamma^2+\sigma_P^2)})$, $m_Q(x) = \sqrt{\frac{\gamma^2}{(\sigma^2+\sigma_{\mathrm{cond}}^2+\gamma^2)}} \exp(-\frac{x^2}{2(\sigma_P^2+\sigma_{\mathrm{cond}}^2+\gamma^2)})$.

**Empirical estimates.** We artificially defined an estimate $\hat{m}_P = \sum_{i=1}^n w_i k_{\mathcal{X}}(\cdot, X_i)$ as follows. First, we generated $n = 100$ samples $X_1, \ldots, X_{100}$ from a uniform distribution on $[-A, A]$ with some $A > 0$ (specified below). We computed the weights $w_1, \ldots, w_n$ by solving an optimization problem

$$\min_{w \in \mathbb{R}^n} \| \sum_{i=1}^n w_i k_{\mathcal{X}}(\cdot, X_i) - m_P \|_{\mathcal{H}}^2 + \lambda \|w\|^2,$$

and then applied normalization so that $\sum_{i=1}^n w_i = 1$. Here $\lambda > 0$ is a regularization constant, which allows us to control the tradeoff between the error $\|\hat{m}_P - m_P\|_{\mathcal{H}_{\mathcal{X}}}^2$ and the quantity $\sum_{i=1}^n w_i^2 = \|w\|^2$. If $\lambda$ is very small, the resulting $\hat{m}_P$ becomes very accurate, i.e., $\|\hat{m}_P - m_P\|_{\mathcal{H}_{\mathcal{X}}}^2$ is small, but has large $\sum_{i=1}^n w_i^2$. If $\lambda$ is large, the error $\|\hat{m}_P - m_P\|_{\mathcal{H}_{\mathcal{X}}}^2$ may not be very small, but $\sum_{i=1}^n w_i^2$ becomes small. This enables us to see how the error $\|\hat{m}_Q - m_Q\|_{\mathcal{H}_{\mathcal{X}}}^2$ changes as we vary these quantities.

**Comparison.** Given $\hat{m}_P = \sum_{i=1}^n w_i k_{\mathcal{X}}(\cdot, X_i)$, we wish to estimate the kernel mean $m_Q$. We compare three estimators:

- woRes: Estimate $m_Q$ without resampling. Generate samples $X_i' \sim p(\cdot|X_i)$ to produce the estimate $\hat{m}_Q = \sum_{i=1}^n w_i k_{\mathcal{X}}(\cdot, X_i')$. This corresponds to the estimator discussed in Section 3.1.
- Res-KH: First apply the resampling algorithm of Algorithm 2 to $\hat{m}_P$, yielding $\bar{X}_1, \ldots, \bar{X}_n$. Then generate $\bar{X}_i' \sim p(\cdot|\bar{X}_i)$ for each $\bar{X}_i$, giving the estimate $\hat{m}_Q = \frac{1}{n}\sum_{i=1}^n k(\cdot, \bar{X}_i')$. This is the estimator discussed in Section 3.3.
- Res-Trunc: Instead of Algorithm 2, first truncate negative weights in $w_1, \ldots, w_n$ to be 0, and apply normalization to make the sum of the weights to be 1. Then apply the multinomial resampling algorithm of particle methods, and estimate $\hat{m}_Q$ as Res-KH.

**Demonstration.** Before starting quantitative comparisons, we demonstrate how the above estimators work. Figure 3.2 shows demonstration results with $A = 1$. First, note that for $\hat{m}_P = \sum_{i=1}^n w_i k(\cdot, X_i)$, samples associated with large weights are located around the mean of $P$, as the standard deviation of $P$ is relatively small $\sigma_P = 0.1$. Note also that some of the weights are negative. In this example, the error of $\hat{m}_P$ is very small $\|m_P - \hat{m}_P\|_{\mathcal{H}_{\mathcal{X}}}^2 = 8.49e-10$, while that of the estimate $\hat{m}_Q$ given by woRes is $\|\hat{m}_Q - m_Q\|_{\mathcal{H}_{\mathcal{X}}}^2 = 0.125$. This shows that even if $\|m_P - \hat{m}_P\|_{\mathcal{H}_{\mathcal{X}}}^2$ is very

small, the resulting $\|\hat{m}_Q - m_Q\|^2_{\mathcal{H}_\mathcal{X}}$ may not be small, as implied by Theorem 1 and the bound (3.5).

We can observe the following. First, Algorithm 2 successfully discarded samples associated with very small weights. Almost all the generated samples $\bar{X}_1, \ldots, \bar{X}_n$ are located in $[-2\sigma_P, 2\sigma_P]$, where $\sigma_P$ is the standard deviation of $P$. The error is $\|\check{m}_P - m_P\|^2_{\mathcal{H}_\mathcal{X}} = 4.74e-5$, which is greater than $\|m_P - \hat{m}_P\|^2_{\mathcal{H}_\mathcal{X}}$. This is due to the additional error caused by the resampling algorithm. Note that the resulting estimate $\check{m}_Q$ is of the error $\|\check{m}_Q - m_Q\|^2_{\mathcal{H}_\mathcal{X}} = 0.00827$. This is much smaller than the estimate $\hat{m}_Q$ by woRes, showing the merit of the resampling algorithm.

Res-Trunc first truncated the negative weights in $w_1, \ldots, w_n$. Let us see the region where the density of $P$ is very small, i.e. the region outside $[-2\sigma_P, 2\sigma_P]$. We can observe that the absolute values of weights are very small in this region. Note that there exist positive and negative weights. These weights maintain balance such that the amounts of positive and negative values are almost the same. Therefore the truncation of the negative weights breaks this balance. As a result, the amount of the positive weights surpasses the amount needed to represent the density of $P$. This can be seen from the histogram for Res-Trunc: some of the samples $\bar{X}_1, \ldots, \bar{X}_n$ generated by Res-Trunc are located in the region where the density of $P$ is very small. Thus the resulting error $\|\check{m}_P - m_P\|^2_{\mathcal{H}_\mathcal{X}} = 0.0538$ is much larger than that of Res-KH. This demonstrates why the resampling algorithm of particle methods is not appropriate for kernel mean embeddings, as discussed in Section 3.2.

**Effects of the sum of squared weights.** The purpose here is to see how the error $\|\hat{m}_Q - m_Q\|^2_{\mathcal{H}_\mathcal{X}}$ changes as we vary the quantity $\sum_{i=1}^n w_i^2$ (recall that the bound (3.5) indicates that $\|\hat{m}_Q - m_Q\|^2_{\mathcal{H}_\mathcal{X}}$ increases as $\sum_{i=1}^n w_i^2$ increases). To this end, we made $\hat{m}_P = \sum_{i=1}^n w_i k_\mathcal{X}(\cdot, X_i)$ for several values of the regularization constant $\lambda$ as described above. For each $\lambda$, we constructed $\hat{m}_P$, and estimated $m_Q$ using each of the three estimators above. We repeated this 20 times for each $\lambda$, and averaged the values of $\|\hat{m}_P - m_P\|^2_{\mathcal{H}_\mathcal{X}}$, $\sum_{i=1}^n w_i^2$ and the errors $\|\hat{m}_Q - m_Q\|^2_{\mathcal{H}_\mathcal{X}}$ by the three estimators. Figure 3.3 shows these results, where the both axes are in the log scale. Here we used $A = 5$ for the support of the uniform distribution.[5] The results are summarized as follows:

- The error of woRes (blue) increases proportionally to the amount of $\sum_{i=1}^n w_i^2$. This matches the bound (3.5).
- The error of Res-KH are not affected by $\sum_{i=1}^n w_i^2$. Rather, it changes in parallel with the error of $\hat{m}_P$. This is explained by the discussions in Section 3.3 on how our resampling algorithm improves the accuracy of the sampling procedure.

[5]This enables us to maintain the values for $\|\hat{m}_P - m_P\|^2_{\mathcal{H}_\mathcal{X}}$ in almost the same amount, while changing the values for $\sum_{i=1}^n w_i^2$.

- Res-Trunc is worse than Res-KH, especially for large $\sum_{i=1}^{n} w_i^2$. This is also explained with the bound (3.8). Here $\check{m}_P$ is the one given by Res-Trunc, so the error $\|\check{m}_P - m_P\|_{\mathcal{H}_\mathcal{X}}$ can be large due to the truncation of negative weights, as shown in the demonstration results. This makes the resulting error $\|\check{m}_Q - m_Q\|_{\mathcal{H}_\mathcal{X}}$ large.

Note that $m_P$ and $m_Q$ are different kernel means, so it can happen that the errors $\|m_Q - \check{m}_Q\|_{\mathcal{H}_\mathcal{X}}$ by Res-KH are less than $\|m_p - \hat{m}_P\|_{\mathcal{H}_\mathcal{X}}$, as in Figure 3.3.

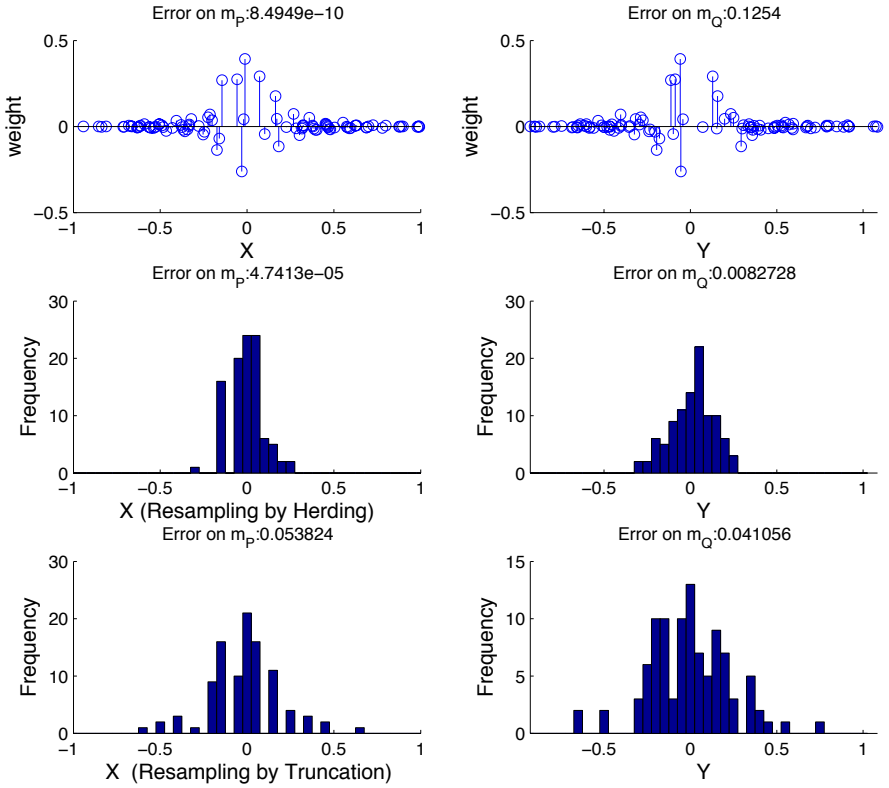



Figure 3.2: Results of the experiments from Section 3.5. Top left and right: sample-weight pairs of $\hat{m}_P = \sum_{i=1}^{n} w_i k_{\mathcal{X}}(\cdot, X_i)$ and $\hat{m}_Q = \sum_{i=1}^{n} w_i k(\cdot, X'_i)$. Middle left and right: histogram of samples $\bar{X}_1, \ldots, \bar{X}_n$ generated by Algorithm 2, and that of samples $\bar{X}'_1, \ldots, \bar{X}'_n$ from the conditional distribution. Bottom left and right: histogram of samples generated with multinomial resampling after truncating negative weights, and that of samples from the conditional distribution.

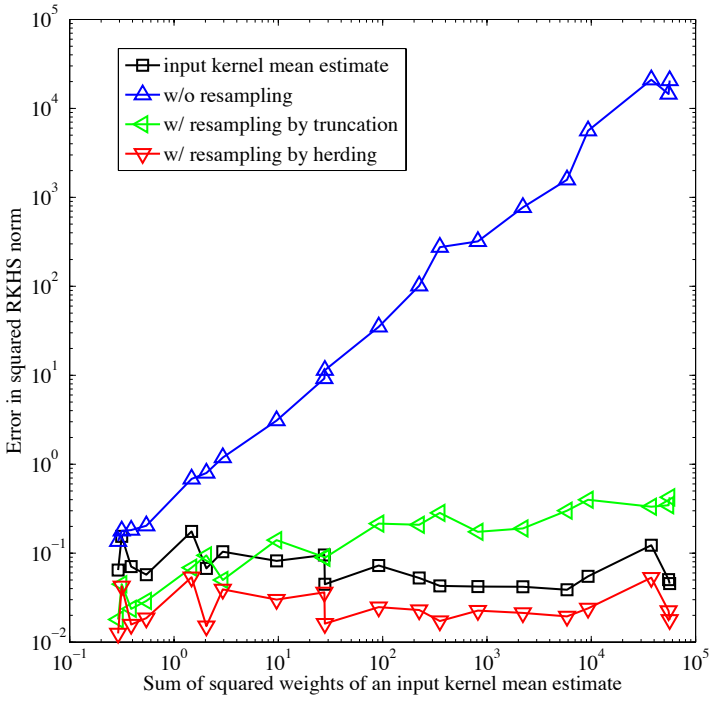



Figure 3.3: Results of synthetic experiments for the sampling and resampling procedure in Section 3.5. Vertical axis: errors in the squared RKHS norm. Horizontal axis: values of $\sum_{i=1}^{n} w_i^2$ for different $\hat{m}_P$. Black: the error of $\hat{m}_P$ ($\|\hat{m}_P - m_P\|_{\mathcal{H}_\mathcal{X}}^2$). Blue, Green and Red: the errors on $m_Q$ by woRes, Res-KH and Res-Trunc, respectively.

## 3.6 Proofs

### 3.6.1 Proof of Theorem 1

Before going to the proof, we review some basic facts that will be needed. Let $m_P = \int k_{\mathcal{X}}(\cdot, x)dP(x)$ and $\hat{m}_P = \sum_{i=1}^n w_i k_{\mathcal{X}}(\cdot, X_i)$. By the reproducing property of the kernel $k_{\mathcal{X}}$, the following hold for any $f \in \mathcal{H}_{\mathcal{X}}$:

$$\begin{aligned} \langle m_P, f\rangle_{\mathcal{H}_{\mathcal{X}}} &= \left\langle \int k_{\mathcal{X}}(\cdot, x)dP(x), f \right\rangle_{\mathcal{H}_{\mathcal{X}}} = \int \langle k_{\mathcal{X}}(\cdot, x), f\rangle_{\mathcal{H}_{\mathcal{X}}} dP(x) \\ &= \int f(x)dP(x) = \mathbf{E}_{X\sim P}[f(X)]. \end{aligned} \tag{3.12}$$

$$\langle \hat{m}_P, f\rangle_{\mathcal{H}_{\mathcal{X}}} = \left\langle \sum_{i=1}^n w_i k_{\mathcal{X}}(\cdot, X_i), f \right\rangle_{\mathcal{H}_{\mathcal{X}}} = \sum_{i=1}^n w_i f(X_i). \tag{3.13}$$

For any $f, g \in \mathcal{H}_{\mathcal{X}}$, we denote by $f \otimes g \in \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}$ the tensor product of $f$ and $g$ defined as

$$f \otimes g(x_1, x_2) := f(x_1)g(x_2) \quad \forall x_1, x_2 \in \mathcal{X}. \tag{3.14}$$

The inner product of the tensor RKHS $\mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}$ satisfies

$$\langle f_1 \otimes g_1, f_2 \otimes g_2\rangle_{\mathcal{H}_{\mathcal{X}}\otimes\mathcal{H}_{\mathcal{X}}} = \langle f_1, f_2\rangle_{\mathcal{H}_{\mathcal{X}}} \langle g_1, g_2\rangle_{\mathcal{H}_{\mathcal{X}}} \quad \forall f_1, f_2, g_1, g_2 \in \mathcal{H}_{\mathcal{X}}. \tag{3.15}$$

Let $\{\phi_i\}_{s=1}^I \subset \mathcal{H}_{\mathcal{X}}$ be complete orthonormal bases of $\mathcal{H}_{\mathcal{X}}$, where $I \in \mathbb{N} \cup \{\infty\}$. Assume $\theta \in \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}$ (recall that this is an assumption of Theorem 1). Then $\theta$ is expressed as

$$\theta = \sum_{s,t=1}^I \alpha_{s,t} \phi_s \otimes \phi_t \tag{3.16}$$

with $\sum_{s,t} |\alpha_{s,t}|^2 < \infty$ (see, e.g., Aronszajn (1950)).

*Proof of Theorem 1.* Recall that $\hat{m}_Q = \sum_{i=1}^n w_i k_{\mathcal{X}}(\cdot, X_i')$, where $X_i' \sim p(\cdot|X_i)$ ($i =$

$1, \ldots, n$). Then

$$\begin{aligned}
& \mathbf{E}_{X'_1,\ldots,X'_n}[\|\hat{m}_Q - m_Q\|^2_{\mathcal{H}_\mathcal{X}}] \\
= \; & \mathbf{E}_{X'_1,\ldots,X'_n}[\langle \hat{m}_Q, \hat{m}_Q \rangle_{\mathcal{H}_\mathcal{X}} - 2\langle \hat{m}_Q, m_Q \rangle_{\mathcal{H}_\mathcal{X}} + \langle m_Q, m_Q \rangle_{\mathcal{H}_\mathcal{X}}] \\
= \; & \sum_{i,j=1}^{n} w_i w_j \mathbf{E}_{X'_i,X'_j}[k_\mathcal{X}(X'_i, X'_j)] \\
& -2\sum_{i=1}^{n} w_i \mathbf{E}_{X'\sim Q, X'_i}[k_\mathcal{X}(X', X'_i)] + \mathbf{E}_{X',\tilde{X}'\sim Q}[k_\mathcal{X}(X', \tilde{X}')] \\
= \; & \sum_{i\neq j} w_i w_j \mathbf{E}_{X'_i,X'_j}[k_\mathcal{X}(X'_i, X'_j)] + \sum_{i=1}^{n} w_i^2 \mathbf{E}_{X'_i}[k_\mathcal{X}(X'_i, X'_i)] \\
& -2\sum_{i=1}^{n} w_i \mathbf{E}_{X'\sim Q, X'_i}[k_\mathcal{X}(X', X'_i)] + \mathbf{E}_{X',\tilde{X}'\sim Q}[k_\mathcal{X}(X', \tilde{X}')], \qquad (3.17)
\end{aligned}$$

where $\tilde{X}'$ denotes an independent copy of $X'$.

Recall that $Q = \int p(\cdot|x)dP(x)$ and $\theta(x, \tilde{x}) := \int\int k_\mathcal{X}(x', \tilde{x}')dp(x'|x)dp(\tilde{x}'|\tilde{x})$. We can then rewrite terms in (3.17) as

$$\begin{aligned}
& \mathbf{E}_{X'\sim Q, X'_i}[k_\mathcal{X}(X', X'_i)] \\
= \; & \int \left( \int\int k_\mathcal{X}(x', x'_i)dp(x'|x)dp(x'_i|X_i) \right) dP(x) \\
= \; & \int \theta(x, X_i)dP(x) = \mathbf{E}_{X\sim P}[\theta(X, X_i)]. \\
& \mathbf{E}_{X',\tilde{X}'\sim Q}[k_\mathcal{X}(X', \tilde{X}')] \\
= \; & \int\int \left( \int\int k_\mathcal{X}(x', \tilde{x}')dp(x'|x)p(\tilde{x}'|\tilde{x}) \right) dP(x)dP(\tilde{x}) \\
= \; & \int\int \theta(x, \tilde{x})dP(x)dP(\tilde{x}) = \mathbf{E}_{X,\tilde{X}\sim P}[\theta(X, \tilde{X})].
\end{aligned}$$

Thus (3.17) is equal to

$$\begin{aligned}
& \sum_{i=1}^{n} w_i^2 \left( \mathbf{E}_{X'_i}[k_\mathcal{X}(X'_i, X'_i)] - \mathbf{E}_{X'_i,\tilde{X}'_i}[k_\mathcal{X}(X'_i, \tilde{X}'_i)] \right) \\
+ \; & \sum_{i,j=1}^{n} w_i w_j \theta(X_i, X_j) - 2\sum_{i=1}^{n} w_i \mathbf{E}_{X\sim P}[\theta(X, X_i)] + \mathbf{E}_{X,\tilde{X}\sim P}[\theta(X, \tilde{X})] \qquad (3.18)
\end{aligned}$$

We can rewrite terms in (3.18) as follows, using the facts (3.12) (3.13) (3.14) (3.15)

(3.16):

$$\begin{aligned}
\sum_{i,j} w_i w_j \theta(X_i, X_j) &= \sum_{i,j} w_i w_j \sum_{s,t} \alpha_{s,t} \phi_s(X_i) \phi_t(X_j) \\
&= \sum_{s,t} \alpha_{s,t} \sum_i w_i \phi_s(X_i) \sum_j w_j \phi_t(X_j) = \sum_{s,t} \alpha_{s,t} \left\langle \hat{m}_P, \phi_s \right\rangle_{\mathcal{H}_\mathcal{X}} \left\langle \hat{m}_P, \phi_t \right\rangle_{\mathcal{H}_\mathcal{X}} \\
&= \sum_{s,t} \alpha_{s,t} \left\langle \hat{m}_P \otimes \hat{m}_P, \phi_s \otimes \phi_t \right\rangle_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}} = \left\langle \hat{m}_P \otimes \hat{m}_P, \theta \right\rangle_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}} . \\
\sum_i w_i \mathbf{E}_{X \sim P}[\theta(X, X_i)] &= \sum_i w_i \mathbf{E}_{X \sim P}[\sum_{s,t} \alpha_{s,t} \phi_s(X) \phi_t(X_i)] \\
&= \sum_{s,t} \alpha_{s,t} \mathbf{E}_{X \sim P}[\phi_s(X)] \sum_i w_i \phi_t(X_i) = \sum_{s,t} \alpha_{s,t} \left\langle m_P, \phi_s \right\rangle_{\mathcal{H}_\mathcal{X}} \left\langle \hat{m}_P, \phi_t \right\rangle_{\mathcal{H}_\mathcal{X}} \\
&= \sum_{s,t} \alpha_{s,t} \left\langle m_P \otimes \hat{m}_P, \phi_s \otimes \phi_t \right\rangle_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}} = \left\langle m_P \otimes \hat{m}_P, \theta \right\rangle_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}} . \\
\mathbf{E}_{X, \tilde{X} \sim P}[\theta(X, \tilde{X})] &= \mathbf{E}_{X, \tilde{X} \sim P}[\sum_{s,t} \alpha_{s,t} \phi_s(X) \phi_t(\tilde{X})] \\
&= \sum_{s,t} \alpha_{s,t} \left\langle m_P, \phi_s \right\rangle_{\mathcal{H}_\mathcal{X}} \left\langle m_P, \phi_t \right\rangle_{\mathcal{H}_\mathcal{X}} = \sum_{s,t} \alpha_{s,t} \left\langle m_P \otimes m_P, \phi_s \otimes \phi_t \right\rangle_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}} \\
&= \left\langle m_P \otimes m_P, \theta \right\rangle_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}} .
\end{aligned}$$

Thus (3.18) is equal to

$$\begin{aligned}
&\sum_{i=1}^n w_i^2 \left( \mathbf{E}_{X_i'}[k_\mathcal{X}(X_i', X_i')] - \mathbf{E}_{X_i', \tilde{X}_i'}[k_\mathcal{X}(X_i', \tilde{X}_i')] \right) \\
&+ \left\langle \hat{m}_P \otimes \hat{m}_P, \theta \right\rangle_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}} - 2 \left\langle \hat{m}_P \otimes m_P, \theta \right\rangle_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}} + \left\langle m_P \otimes m_P, \theta \right\rangle_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}} \\
= \quad &\sum_{i=1}^n w_i^2 \left( \mathbf{E}_{X_i'}[k_\mathcal{X}(X_i', X_i')] - \mathbf{E}_{X_i', \tilde{X}_i'}[k_\mathcal{X}(X_i', \tilde{X}_i')] \right) \\
&+ \left\langle (\hat{m}_P - m_P) \otimes (\hat{m}_P - m_P), \theta \right\rangle_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}} .
\end{aligned}$$

Finally, the Cauchy-Schwartz inequality gives

$$\left\langle (\hat{m}_P - m_P) \otimes (\hat{m}_P - m_P), \theta \right\rangle_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}} \leq \|\hat{m}_P - m_P\|_{\mathcal{H}_\mathcal{X}}^2 \|\theta\|_{\mathcal{H}_\mathcal{X} \otimes \mathcal{H}_\mathcal{X}}.$$

This completes the proof. □

### 3.6.2 Proof of Theorem 2

Theorem 2 provides convergence rates for the resampling algorithm (Algorithm 3). This theorem assumes that the candidate samples $Z_1, \ldots, Z_N$ for resampling are i.i.d. with a density $q$. Here we prove Theorem 2 by showing that the same statement holds under weaker assumptions (Theorem 3 below).

We first describe assumptions. Let $P$ be the distribution of the kernel mean $m_P$, and $L_2(P)$ be the Hilbert space of square-integrable functions on $\mathcal{X}$ with respect to $P$. For any $f \in L_2(P)$, we write its norm by $\|f\|_{L_2(P)} := \int f^2(x)dP(x)$.

**Assumption 1.** *The candidate samples $Z_1, \ldots, Z_N$ are independent. There are probability distributions $Q_1, \ldots, Q_N$ on $\mathcal{X}$, such that for any bounded measurable function $g : \mathcal{X} \to \mathbb{R}$, we have*

$$\mathbf{E}\left[\frac{1}{N-1}\sum_{j\neq i} g(Z_j)\right] = \mathbf{E}_{X\sim Q_i}[g(X)] \quad (i = 1, \ldots, N). \tag{3.19}$$

**Assumption 2.** *The distributions $Q_1, \ldots, Q_N$ have density functions $q_1, \ldots, q_N$, respectively. Define $Q := \frac{1}{N}\sum_{i=1}^N Q_i$ and $q := \frac{1}{N}\sum_{i=1}^N q_i$. There is a constant $A > 0$ that does not depend on $N$, such that*

$$\left\|\frac{q_i}{q} - 1\right\|^2_{L_2(P)} \leq \frac{A}{\sqrt{N}} \quad (i = 1, \ldots, N). \tag{3.20}$$

**Assumption 3.** *The distribution $P$ has a density function $p$ such that $\sup_{x\in\mathcal{X}} \frac{p(x)}{q(x)} < \infty$. There is a constant $\sigma > 0$ such that*

$$\sqrt{N}\left(\frac{1}{N}\sum_{i=1}^N \frac{p(Z_i)}{q(Z_i)} - 1\right) \xrightarrow{D} \mathcal{N}(0, \sigma^2), \tag{3.21}$$

*where $\xrightarrow{D}$ denotes convergence in distribution and $\mathcal{N}(0, \sigma^2)$ the normal distribution with mean 0 and variance $\sigma^2$.*

These assumptions are weaker than those in Theorem 2, which require $Z_1, \ldots, Z_N$ be i.i.d. For example, Assumption 1 is clearly satisfied for the i.i.d. case, since in this case we have $Q = Q_1, = \cdots = Q_N$. The inequality (3.20) in Assumption 2 requires that the distributions $Q_1, \ldots, Q_N$ get similar, as the sample size increases. This is also satisfied under the i.i.d. assumption. Likewise, the convergence (3.21) in Assumption 3 is satisfied from the central limit theorem if $Z_1, \ldots, Z_N$ are i.i.d.

We will need the following lemma.

**Lemma 1.** *Let $Z_1, \ldots, Z_N$ be samples satisfying Assumption 1. Then the following holds for any bounded measurable function $g : \mathcal{X} \to \mathbb{R}$:*

$$\mathbf{E}\left[\frac{1}{N}\sum_{i=1}^{N} g(Z_i)\right] = \int g(x)dQ(x).$$

*Proof.*

$$\begin{aligned}\mathbf{E}\left[\frac{1}{N}\sum_{i=1}^{N} g(Z_i)\right] &= \mathbf{E}\left[\frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j\neq i} g(Z_j)\right] \\ = \frac{1}{N}\sum_{i=1}^{N}\mathbf{E}\left[\frac{1}{N-1}\sum_{j\neq i} g(Z_j)\right] &= \frac{1}{N}\sum_{i=1}^{N}\int g(x)Q_i(x) = \int g(x)dQ(x).\end{aligned}$$

□

The following theorem shows the convergence rates of our resampling algorithm. Note that it does not assume that the candidate samples $Z_1, \ldots, Z_N$ are identical to those expressing the estimator $\hat{m}_P$.

**Theorem 3.** *Let $k$ be a bounded positive definite kernel, and $\mathcal{H}$ be the associated RKHS. Let $Z_1, \ldots, Z_N$ be candidate samples satisfying Assumptions 1, 2 and 3. Let $P$ be a probability distribution satisfying Assumption 3, and let $m_P = \int k(\cdot, x)dP(x)$ be the kernel mean. Let $\hat{m}_P \in \mathcal{H}$ be any element in $\mathcal{H}$. Suppose we apply Algorithm 3 to $\hat{m}_P \in \mathcal{H}$ with candidate samples $Z_1, \ldots, Z_N$, and let $\bar{X}_1, ..., \bar{X}_\ell \in \{Z_1, \ldots, Z_N\}$ be the resulting samples. Then the following holds:*

$$\left\| m_P - \frac{1}{\ell}\sum_{i=1}^{\ell} k(\cdot, \bar{X}_i)\right\|_{\mathcal{H}}^2 = \left(\|\hat{m}_P - m_P\|_{\mathcal{H}_\mathcal{X}} + O_p(N^{-1/2})\right)^2 + O\left(\frac{\ln \ell}{\ell}\right).$$

*Proof.* Our proof is based on the fact (Bach et al., 2012) that Kernel Herding can be seen as the Frank-Wolfe optimization method with step size $1/(\ell+1)$ for the $\ell$-th iteration. For details of the Frank-Wolfe method, we refer to Jaggi (2013); Freund and Grigas (2014) and references therein.

Fix the samples $Z_1, \ldots, Z_N$. Let $\mathcal{M}_N$ be the convex hull of the set $\{k(\cdot, Z_1), \ldots, k(\cdot, Z_N)\} \subset \mathcal{H}$. Define a loss function $J : \mathcal{H} \to \mathbb{R}$ by

$$J(g) = \frac{1}{2}\|g - \hat{m}_P\|_{\mathcal{H}}^2, \quad g \in \mathcal{H} \tag{3.22}$$

Then Algorithm 3 can be seen as the Frank-Wolfe method that iteratively minimizes

this loss function over the convex hull $\mathcal{M}_N$:

$$\inf_{g \in \mathcal{M}_N} J(g).$$

More precisely, the Frank-Wolfe method solves this problem by the following iterations:

$$\begin{aligned} s &:= \arg\min_{g \in \mathcal{M}_N} \langle g, \nabla J(g_{\ell-1}) \rangle_{\mathcal{H}} \\ g_\ell &:= (1-\gamma) g_{\ell-1} + \gamma s \quad (\ell \geq 1), \end{aligned}$$

where $\gamma$ is a step size defined as $\gamma = 1/\ell$, and $\nabla J(g_{\ell-1})$ is the gradient of $J$ at $g_{\ell-1}$: $\nabla J(g_{\ell-1}) = g_{\ell-1} - \hat{m}_P$. Here the initial point is defined as $g_0 := 0$. It can be easily shown that $g_\ell = \frac{1}{\ell} \sum_{i=1}^{\ell} k(\cdot, \bar{X}_i)$, where $\bar{X}_1, \ldots, \bar{X}_\ell$ are the samples given by Algorithm 3. For details, see Bach et al. (2012).

Let $L_{J,\mathcal{M}_N} > 0$ be the Lipschitz constant of the gradient $\nabla J$ over $\mathcal{M}_N$, and Diam $\mathcal{M}_N > 0$ be the diameter of $\mathcal{M}_N$:

$$\begin{aligned} L_{J,\mathcal{M}_N} &:= \sup_{g_1, g_2 \in \mathcal{M}_N} \frac{\|\nabla J(g_1) - \nabla J(g_2)\|_{\mathcal{H}}}{\|g_1 - g_2\|_{\mathcal{H}}} \\ &= \sup_{g_1, g_2 \in \mathcal{M}_N} \frac{\|g_1 - g_2\|_{\mathcal{H}}}{\|g_1 - g_2\|_{\mathcal{H}}} = 1, \qquad (3.23) \\ \text{Diam } \mathcal{M}_N &:= \sup_{g_1, g_2 \in \mathcal{M}_N} \|g_1 - g_2\|_{\mathcal{H}} \\ &\leq \sup_{g_1, g_2 \in \mathcal{M}_N} \|g_1\|_{\mathcal{H}} + \|g_2\|_{\mathcal{H}} \leq 2C, \qquad (3.24) \end{aligned}$$

where $C := \sup_{x \in \mathcal{X}} \|k(\cdot, x)\|_{\mathcal{H}} = \sup_{x \in \mathcal{X}} \sqrt{k(x,x)} < \infty$.

From Bound 3.2 and Eq. (8) of Freund and Grigas (2014), we then have

$$\begin{aligned} J(g_\ell) - \inf_{g \in \mathcal{M}_N} J(g) &\leq \frac{L_{J,\mathcal{M}_N} (\text{Diam } \mathcal{M}_N)^2 (1 + \ln \ell)}{2\ell} \qquad (3.25) \\ &\leq \frac{2C^2(1 + \ln \ell)}{\ell}, \qquad (3.26) \end{aligned}$$

where the last inequality follows from (3.23) and (3.24).

Note that the upper-bound of (3.26) does not depend on the candidate samples $Z_1, \ldots, Z_N$. Hence, combined with (3.22), the following holds for any choice of

$Z_1, \ldots, Z_N$:

$$\left\| \hat{m}_P - \frac{1}{\ell} \sum_{i=1}^{\ell} k(\cdot, \bar{X}_i) \right\|_{\mathcal{H}}^2 \leq \inf_{g \in \mathcal{M}_N} \|\hat{m}_P - g\|_{\mathcal{H}}^2 + \frac{4C^2(1 + \ln \ell)}{\ell}. \tag{3.27}$$

Below we will focus on bounding the first term of (3.27). Recall here that $Z_1, \ldots, Z_N$ are random samples. Define a random variable $S_N := \sum_{i=1}^{N} \frac{p(Z_i)}{q(Z_i)}$. Since $\mathcal{M}_N$ is the convex hull of the $\{k(\cdot, Z_1), \ldots, k(\cdot, Z_N)\}$, we have

$$\begin{aligned}
& \inf_{g \in \mathcal{M}_N} \|\hat{m}_P - g\|_{\mathcal{H}} \\
= & \inf_{\alpha \in \mathbb{R}^N,\ \alpha \geq 0,\ \sum_i \alpha_i \leq 1} \|\hat{m}_P - \sum_i \alpha_i k(\cdot, Z_i)\|_{\mathcal{H}} \\
\leq & \|\hat{m}_P - \frac{1}{S_N} \sum_i \frac{p(Z_i)}{q(Z_i)} k(\cdot, Z_i)\|_{\mathcal{H}} \\
\leq & \|\hat{m}_P - m_P\|_{\mathcal{H}} + \|m_P - \frac{1}{N} \sum_i \frac{p(Z_i)}{q(Z_i)} k(\cdot, Z_i)\|_{\mathcal{H}} \\
& + \|\frac{1}{N} \sum_i \frac{p(Z_i)}{q(Z_i)} k(\cdot, Z_i) - \frac{1}{S_N} \sum_i \frac{p(Z_i)}{q(Z_i)} k(\cdot, Z_i)\|_{\mathcal{H}}.
\end{aligned}$$

Therefore we have

$$\begin{aligned}
& \|\hat{m}_P - \frac{1}{\ell} \sum_{i=1}^{\ell} k(\cdot, \bar{X}_i)\|_{\mathcal{H}}^2 \\
\leq & (\|\hat{m}_P - m_P\|_{\mathcal{H}} + \|m_P - \frac{1}{N} \sum_i \frac{p(Z_i)}{q(Z_i)} k(\cdot, Z_i)\|_{\mathcal{H}} \\
& + \|\frac{1}{N} \sum_i \frac{p(Z_i)}{q(Z_i)} k(\cdot, Z_i) - \frac{1}{S_N} \sum_i \frac{p(Z_i)}{q(Z_i)} k(\cdot, Z_i)\|_{\mathcal{H}})^2 + O\left(\frac{\ln \ell}{\ell}\right).
\end{aligned} \tag{3.28}$$

Below we derive rates of convergence for the second and third terms.

**Second term.** We derive a rate of convergence in expectation, which implies a rate of convergence in probability. To this end, we use the following fact: Let $f \in \mathcal{H}$ be any function in the RKHS. By the assumption $\sup_{x \in \mathcal{X}} \frac{p(x)}{q(x)} < \infty$ and the boundedness

of $k$, functions $x \to \frac{p(x)}{q(x)} f(x)$ and $x \to \left(\frac{p(x)}{q(x)}\right)^2 f(x)$ are bounded.

$$
\begin{aligned}
& \mathbf{E}[\|m_P - \frac{1}{N}\sum_i \frac{p(Z_i)}{q(Z_i)} k(\cdot, Z_i)\|_{\mathcal{H}}^2] \\
= \; & \|m_P\|_{\mathcal{H}}^2 - 2\mathbf{E}[\frac{1}{N}\sum_i \frac{p(Z_i)}{q(Z_i)} m_P(Z_i)] + \mathbf{E}[\frac{1}{N^2}\sum_i \sum_j \frac{p(Z_i)}{q(Z_i)} \frac{p(Z_j)}{q(Z_j)} k(Z_i, Z_j)] \\
= \; & \|m_P\|_{\mathcal{H}}^2 - 2\int \frac{p(x)}{q(x)} m_P(x) q(x) dx + \mathbf{E}[\frac{1}{N^2}\sum_i \sum_{j \neq i} \frac{p(Z_i)}{q(Z_i)} \frac{p(Z_j)}{q(Z_j)} k(Z_i, Z_j)] \\
& + \mathbf{E}[\frac{1}{N^2}\sum_i \left(\frac{p(Z_i)}{q(Z_i)}\right)^2 k(Z_i, Z_i)] \\
= \; & \|m_P\|_{\mathcal{H}}^2 - 2\|m_P\|_{\mathcal{H}}^2 + \mathbf{E}[\frac{N-1}{N^2}\sum_i \frac{p(Z_i)}{q(Z_i)} \int \frac{p(x)}{q(x)} k(Z_i, x) q_i(x) dx] \\
& + \frac{1}{N}\int \left(\frac{p(x)}{q(x)}\right)^2 k(x, x) q(x) dx \\
= \; & -\|m_P\|_{\mathcal{H}}^2 + \mathbf{E}[\frac{N-1}{N^2}\sum_i \frac{p(Z_i)}{q(Z_i)} \int \frac{p(x)}{q(x)} k(Z_i, x) q_i(x) dx] + \frac{1}{N}\int \frac{p(x)}{q(x)} k(x, x) dP(x).
\end{aligned}
$$

We further rewrite the second term of the last equality as follows:

$$
\begin{aligned}
& \mathbf{E}[\frac{N-1}{N^2}\sum_i \frac{p(Z_i)}{q(Z_i)} \int \frac{p(x)}{q(x)} k(Z_i, x) q_i(x) dx] \\
= \; & \mathbf{E}[\frac{N-1}{N^2}\sum_i \frac{p(Z_i)}{q(Z_i)} \int \frac{p(x)}{q(x)} k(Z_i, x)(q_i(x) - q(x)) dx] \\
& + \mathbf{E}[\frac{N-1}{N^2}\sum_i \frac{p(Z_i)}{q(Z_i)} \int \frac{p(x)}{q(x)} k(Z_i, x) q(x) dx] \\
= \; & \mathbf{E}[\frac{N-1}{N^2}\sum_i \frac{p(Z_i)}{q(Z_i)} \int \sqrt{p(x)} k(Z_i, x) \sqrt{p(x)} (\frac{q_i(x)}{q(x)} - 1) dx] + \frac{N-1}{N}\|m_P\|_{\mathcal{H}}^2 \\
\leq \; & \mathbf{E}[\frac{N-1}{N^2}\sum_i \frac{p(Z_i)}{q(Z_i)} \|k(Z_i, \cdot)\|_{L_2(P)} \|\frac{q_i(x)}{q(x)} - 1\|_{L_2(P)}] + \frac{N-1}{N}\|m_P\|_{\mathcal{H}}^2 \\
\leq \; & \mathbf{E}[\frac{N-1}{N^3}\sum_i \frac{p(Z_i)}{q(Z_i)} C^2 A] + \frac{N-1}{N}\|m_P\|_{\mathcal{H}}^2 \\
= \; & \frac{C^2 A(N-1)}{N^2} + \frac{N-1}{N}\|m_P\|_{\mathcal{H}}^2,
\end{aligned}
$$

where the first inequality follows from Cauchy-Schwartz. Using this, we obtain

$$\begin{aligned} & \mathbf{E}[\|m_P - \frac{1}{N}\sum_i \frac{p(Z_i)}{q(Z_i)} k(\cdot, Z_i)\|_{\mathcal{H}}^2 \\ \leq & \frac{1}{N}\left(\int \frac{p(x)}{q(x)} k(x,x) dP(x) - \|m_P\|_{\mathcal{H}}^2\right) + \frac{C^2(N-1)A}{N^2} \\ = & O(N^{-1}). \end{aligned}$$

Therefore we have

$$\|m_P - \frac{1}{N}\sum_i \frac{p(Z_i)}{q(Z_i)} k(\cdot, Z_i)\|_{\mathcal{H}} = O_p(N^{-1/2}) \quad (N \to \infty). \tag{3.29}$$

**Third term.** We can bound the third term as follows:

$$\begin{aligned} & \left\| \frac{1}{N}\sum_i \frac{p(Z_i)}{q(Z_i)} k(\cdot, Z_i) - \frac{1}{S_N}\sum_i \frac{p(Z_i)}{q(Z_i)} k(\cdot, Z_i) \right\|_{\mathcal{H}} \\ = & \left\| \frac{1}{N}\sum_i \frac{p(Z_i)}{q(Z_i)} k(\cdot, Z_i) \left(1 - \frac{N}{S_N}\right) \right\|_{\mathcal{H}} \\ = & \left| 1 - \frac{N}{S_N} \right| \left\| \frac{1}{N}\sum_i \frac{p(Z_i)}{q(Z_i)} k(\cdot, Z_i) \right\|_{\mathcal{H}} \\ \leq & \left| 1 - \frac{N}{S_N} \right| C \ \|p/q\|_\infty \\ = & \left| 1 - \frac{1}{\frac{1}{N}\sum_{i=1}^N p(Z_i)/q(Z_i)} \right| C \ \|p/q\|_\infty, \end{aligned}$$

where $\|p/q\|_\infty := \sup_{x \in \mathcal{X}} \frac{p(x)}{q(x)} < \infty$. Therefore the following holds by Assumption 3 and the Delta method:

$$\left\| \frac{1}{N}\sum_i \frac{p(Z_i)}{q(Z_i)} k(\cdot, Z_i) - \frac{1}{S_N}\sum_i \frac{p(Z_i)}{q(Z_i)} k(\cdot, Z_i) \right\|_{\mathcal{H}} = O_p(N^{-1/2}). \tag{3.30}$$

The assertion of the theorem follows from (3.28) (3.29) (3.30). □

# Chapter 4

# Kernel Monte Carlo Filter

Time-series data are ubiquitous in science and engineering. We often wish to extract useful information from such time-series data. *State-space models* have been one of the most successful approaches for this purpose (see, e.g., Durbin and Koopman (2012)). Suppose that we have a sequence of observations $y_1, \ldots, y_t, \ldots, y_T$. A state-space model assumes that for each observation $y_t$, there is a hidden state $x_t$ that generates $y_t$, and that these states $x_1, \ldots, x_t, \ldots, x_T$ follow a Markov process (see Figure 4.1). Therefore the state-space model is characterized by two components: (1) *observation model* $p(y_t|x_t)$, the conditional distribution of an observation given a state, and (2) *transition model* $p(x_t|x_{t-1})$, the conditional distribution of a state given the previous one.

This chapter addresses the problem of *filtering*, which has been a central topic in the literature on state-space models. The task is to estimate a posterior distribution of the state for each time $t$, based on observations up to that time:

$$p(x_t|y_1, \ldots, y_t), \quad t = 1, 2, \ldots, T. \tag{4.1}$$

The estimation is to be done online (sequentially), as each $y_t$ is received. For example, a tracking problem can be formulated as filtering, where $x_t$ is the position of an object to be tracked, and $y_t$ is a noisy observation of $x_t$ (Ristic et al., 2004).

As an inference problem, the starting point of filtering is that the observation model $p(y_t|x_t)$ and the transition model $p(x_t|x_{t-1})$ are *given* in some form. The simplest form is a liner-Gaussian state-space model, which enables analytic computation of the posteriors; this is the principle of the classical Kalman filter (Kalman, 1960). The filtering problem is more difficult if the observation and transition models involve nonlinear-transformation and non-Gaussian noise. Standard solutions for such situations include Extended and Unscented Kalman filters (Anderson and Moore, 1979; Julier and Uhlmann, 1997, 2004) and particle filters (Gordon et al., 1993; Doucet et al., 2001; Doucet and Johansen, 2011). Particle filters in particular have wide

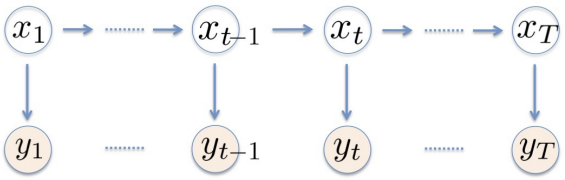



Figure 4.1: Graphical representation of a state-space model: $y_1, \ldots, y_T$ denote observations, and $x_1, \ldots, x_T$ denote states. The states are hidden, and to be estimated from the observations.

applicability since they only require that (i) (unnormalized) density values of the observation model are computable, and that (ii) sampling with the transition model is possible. Thus particle methods are applicable to basically any nonlinear non-Gaussian state-space models, and have been used in various fields such as computer vision, robotics, computational biology, and so on (see, e.g., Doucet et al. (2001)).

However, it can even be restrictive to assume that the observation model $p(y_t|x_t)$ is given as a probabilistic model. An important point is that in practice, we may define the states $x_1, \ldots, x_T$ arbitrarily as quantities that we wish to estimate from available observations $y_1, \ldots, y_T$. Thus if these quantities are very different from the observations, the observation model may not admit a simple parametric form. For example, in location estimation problems in robotics, states are locations in a map, while observations are sensor data, such as camera images and signal strength measurements of a wireless device (Vlassis et al., 2002; Wolf et al., 2005; Ferris et al., 2006). In brain computer interface applications, states are defined as positions of a device to be manipulated, while observations are brain signals (Pistohl et al., 2008; Wang et al., 2011). In these applications, it is hard to define the observation model as a probabilistic model in parametric form.

For such applications where the observation model is very complicated, information about the relation between states and observations is rather given as *examples* of state-observation pairs $\{(X_i, Y_i)\}$; such examples are often available *before* conducting filtering in test phase. For example, one can collect location-sensor examples for the location estimation problems, by making use of more expensive sensors than those for filtering (Quigley et al., 2010). The brain computer interface problems also allow us to obtain training samples for the relation between device positions and brain signals (Schalk et al., 2007). However, making use of such examples for learning the observation model is not straightforward. If one relies on a parametric approach, it would require exhaustive efforts for designing a parametric model to fit the complicated (true) observation model. Nonparametric methods such as kernel density estimation (Silverman, 1986), on the other hand, suffer from the curse of dimensionality when

applied to high-dimensional observations. Moreover, observations may be suitable to be represented as *structured* (non-vectorial) data, as for the cases of image and text. Such situations are not straightforward for either approach, since they usually require that data is given as real vectors.

We propose a filtering method that is focused on the above situations where the information of the observation model $p(y_t|x_t)$ is only given through the state-observation examples $\{(X_i, Y_i)\}$: we do not assume any parametric model for the observation model. On the other hand, we assume that the transition model is known, as for a standard particle filter: the probabilistic model can be arbitrarily nonlinear and non-Gaussian.

We develop a filtering method for this setting based on kernel mean embeddings. We call it *Kernel Monte Carlo Filter (KMCF)*. As it is based on kernel mean embeddings, all the involved distributions are expressed as kernel means. The filtering problem can then be cast as how to estiamte the kernel means of the posterior distributions. Specifically, estimation of each posterior kernel mean is done by the combination of Kernel Bayes' Rule in Section 2.5.2 and the sampling and resampling procedures developed in Chapter 3. More precisely, this estimation consists of three steps of *prediction*, *correction* and *resampling*. Suppose that we already obtained an estimate for the posterior of the previous time. In the prediction step, this previous estimate is propagated forward by sampling with the transition model. The propagated estimate is then used as a prior for the current state. In the correction step, Kernel Bayes' Rule is applied to obtain a posterior estimate, using the prior and the state-observation examples $\{(X_i, Y_i)\}_{i=1}^n$. Finally, in the resampling step, pseudo samples are obtained from the posterior estimate by applying the resampling algorithm. These samples are then used in the prediction step of the next iteration. We show that this algorithm is consistent: KMCF provide posterior estimates that approach to the true posteriors, as the number of state-observation examples $\{(X_i, Y_i)\}_{i=1}^n$ increases.

This chapter proceeds as follows. We first review related works in Section 4.1. We then present the KMCF algorithm in Section 4.2, and show how it can be accelerated in Section 4.3. We show consistency of KMCF in Section 4.4, and finally report experimental results in Section 4.5.

## 4.1 Related work

As explained, we consider the following setting: (i) the observation model $p(y_t|x_t)$ is not known explicitly or even parametrically. Instead, state-observation examples $\{(X_i, Y_i)\}$ are available before test phase; (ii) sampling from the transition model $p(x_t|x_{t-1})$ is possible. Note that standard particle filters cannot be applied to this setting directly, since they require that the observation model is given as a parametric model.

As far as we know, there exist a few methods that can be applied to this setting directly (Vlassis et al., 2002; Ferris et al., 2006). These methods learn the observation model from state-observation examples nonparametrically, and then use it to run a particle filter with a transition model. Vlassis et al. (2002) proposed to apply conditional density estimation based on the $k$-nearest neighbors approach (Stone, 1977) for learning the observation model. A problem here is that conditional density estimation suffers from the curse of dimensionality if observations are high-dimensional (Silverman, 1986). Vlassis et al. (2002) avoided this problem by estimating the conditional density function of the state given observation, and used it as an alternative for the observation model. This heuristic may introduce bias in estimation, however. Ferris et al. (2006) proposed to use Gaussian Process regression for leaning the observation model. This method will perform well if the Gaussian noise assumption is satisfied, but cannot be applied to structured observations.

## 4.2 Proposed method

In this section, we present our Kernel Monte Carlo Filter (KMCF). First, we define notation and review the problem setting in Section 4.2.1. We then describe the algorithm of KMCF in Section 4.2.2. We discuss implementation issues such as hyperparameter selection and computational cost in Section 4.2.3. We explain how to decode the information on the posteriors from the estimated kernel means in Section 4.2.4.

### 4.2.1 Notation and problem setup

Here we formally define the problem. The notation is summarized in Table 4.1.

We consider a state-space model (see Figure 4.1). Let $\mathcal{X}$ and $\mathcal{Y}$ be measurable spaces, which serve as a state space and an observation space, respectively. Let $x_1, \ldots, x_t, \ldots, x_T \in \mathcal{X}$ be a sequence of hidden states, which follow a Markov process. Let $p(x_t|x_{t-1})$ denote a transition model that defines this Markov process. Let $y_1, \ldots, y_t, \ldots, y_T \in \mathcal{Y}$ be a sequence of observations. Each observation $y_t$ is assumed to be generated from an observation model $p(y_t|x_t)$ conditioned on the corresponding state $x_t$. We use the abbreviation $y_{1:t} := y_1, \ldots, y_t$.

We consider a filtering problem of estimating the posterior distribution $p(x_t|y_{1:t})$ for each time $t = 1, \ldots, T$. The estimation is to be done online, as each $y_t$ is given. Specifically, we consider the following setting:

1. The observation model $p(y_t|x_t)$ is not known explicitly, or even parametrically. Instead, we are given examples of state-observation pairs $\{(X_i, Y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ prior to the test phase. The observation model is also assumed time-invariant.

Table 4.1: Notation

| | |
|---|---|
| $\mathcal{X}$ | State space |
| $\mathcal{Y}$ | Observation space |
| $x_t \in \mathcal{X}$ | State at time $t$ |
| $y_t \in \mathcal{Y}$ | Observation at time $t$ |
| $p(y_t\|x_t)$ | Observation model |
| $p(x_t\|x_{t-1})$ | Transition model |
| $\{(X_i, Y_i)\}_{i=1}^n$ | State-observation examples |
| $k_{\mathcal{X}}$ | Positive definite kernel on $\mathcal{X}$ |
| $k_{\mathcal{Y}}$ | Positive definite kernel on $\mathcal{Y}$ |
| $\mathcal{H}_{\mathcal{X}}$ | RKHS associated with $k_{\mathcal{X}}$ |
| $\mathcal{H}_{\mathcal{Y}}$ | RKHS associated with $k_{\mathcal{Y}}$ |

2. Sampling from the transition model $p(x_t|x_{t-1})$ is possible. Its probabilistic model can be an arbitrary nonlinear non-Gaussian distribution, as for standard particle filters. It can further depend on time. For example, control input can be included in the transition model as $p(x_t|x_{t-1}) := p(x_t|x_{t-1}, u_t)$, where $u_t$ denotes control input provided by a user at time $t$.

Let $k_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be positive definite kernels on $\mathcal{X}$ and $\mathcal{Y}$, respectively. Denote by $\mathcal{H}_{\mathcal{X}}$ and $\mathcal{H}_{\mathcal{Y}}$ their respective RKHSs. We address the above filtering problem by estimating the **kernel means of the posteriors**:

$$m_{x_t|y_{1:t}} := \int k_{\mathcal{X}}(\cdot, x_t) p(x_t|y_{1:t}) dx_t \in \mathcal{H}_{\mathcal{X}} \quad (t = 1, \ldots, T). \tag{4.2}$$

These preserve all the information of the corresponding posteriors, if the kernels are characteristic (see Section 2.2). Therefore the resulting estimates of these kernel means provide us the information of the posteriors, as explained in Section 4.2.4

### 4.2.2 Algorithm

KMCF iterates three steps of *prediction*, *correction* and *resampling* for each time $t$. Suppose that we have just finished the iteration at time $t-1$. Then, as shown later, the resampling step yields the following estimator of (4.2) at time $t-1$:

$$\check{m}_{x_{t-1}|y_{1:t-1}} := \frac{1}{n} \sum_{i=1}^{n} k_{\mathcal{X}}(\cdot, \bar{X}_{t-1,i}), \tag{4.3}$$

where $\bar{X}_{t-1,1}, \ldots, \bar{X}_{t-1,n} \in \mathcal{X}$. Below we show one iteration of KMCF that estimates the kernel mean (4.2) at time $t$ (see also Figure 4.2).

**1. Prediction step** The prediction step is application of the sampling procedure analyzed in Section 3.1. We generate a sample from the transition model for each $\bar{X}_{t-1,i}$ in (4.3);

$$X_{t,i} \sim p(x_t | x_{t-1} = \bar{X}_{t-1,i}), \quad (i = 1, \ldots, n). \tag{4.4}$$

We then specify a new empirical kernel mean;

$$\hat{m}_{x_t|y_{1:t-1}} := \frac{1}{n} \sum_{i=1}^{n} k_{\mathcal{X}}(\cdot, X_{t,i}). \tag{4.5}$$

This is an estimator of the following kernel mean of the prior;

$$m_{x_t|y_{1:t-1}} := \int k_{\mathcal{X}}(\cdot, x_t) p(x_t | y_{1:t-1}) dx_t \in \mathcal{H}_{\mathcal{X}}, \tag{4.6}$$

where

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

is the prior distribution of the current state $x_t$. Thus (4.5) serves as a prior for the subsequent posterior estimation.

**2. Correction step** This step estimates the kernel mean (4.2) of the posterior by using Kernel Bayes' Rule (Algorithm 1) in Section 2.5.2. This makes use of the new observation $y_t$, the state-observation examples $\{(X_i, Y_i)\}_{i=1}^n$ and the estimate (4.5) of the prior.

The input of Algorithm 1 consists of (i) vectors

$$\begin{aligned} \mathbf{k}_Y &= (k_{\mathcal{Y}}(y_t, Y_1), \ldots, k_{\mathcal{Y}}(y_t, Y_n))^T \in \mathbb{R}^n \\ \mathbf{m}_\pi &= (\hat{m}_{x_t|y_{1:t-1}}(X_1), \ldots, \hat{m}_{x_t|y_{1:t-1}}(X_n))^T \\ &= \left( \frac{1}{n} \sum_{i=1}^{n} k_{\mathcal{X}}(X_q, X_{t,i}) \right)_{q=1}^{n} \in \mathbb{R}^n, \end{aligned}$$

which are interpreted as expressions of $y_t$ and $\hat{m}_{x_t|y_{1:t-1}}$ using the sample $\{(X_i, Y_i)\}_{i=1}^n$, (ii) kernel matrices $G_X = (k_{\mathcal{X}}(X_i, X_j))$, $G_Y = (k_{\mathcal{Y}}(Y_i, Y_j)) \in \mathbb{R}^{n \times n}$, and (iii) regularization constants $\varepsilon, \delta > 0$. These constants $\varepsilon, \delta$ as well as kernels $k_{\mathcal{X}}, k_{\mathcal{Y}}$ are hyper-parameters of KMCF; we will discuss how to choose these parameters later.

Algorithm 1 outputs a weight vector $w := (w_1, \ldots, w_n) \in \mathbb{R}^n$. Normalizing these

weights[1] $w_t := w/\sum_{i=1}^n w_i$, we obtain an estimator of (4.2) as

$$\hat{m}_{x_t|y_{1:t}} = \sum_{i=1}^n w_{t,i} k_{\mathcal{X}}(\cdot, X_i). \tag{4.7}$$

The apparent difference from a particle filter is that the posterior (kernel mean) estimator (4.7) is expressed in terms of the samples $X_1, \ldots, X_n$ in the training sample $\{(X_i, Y_i)\}_{i=1}^n$, not with the samples from the prior (4.5). This requires that the training samples $X_1, \ldots, X_n$ cover the support of posterior $p(x_t|y_{1:t})$ sufficiently well. If this does not hold, we cannot expect good performance for the posterior estimate. Note that this is also true for any methods that deal with the setting of this chapter; poverty of training samples in a certain region means that we do not have any information about the observation model $p(y_t|x_t)$ in that region.

**3. Resampling step** In this step, we apply the resampling algorithm (Algorithm 2) to the empirical kernel mean (4.7). Let $\bar{X}_{t,1}, \ldots, \bar{X}_{t,n}$ be the resulting pseudo samples, and define a new empirical kernel mean

$$\check{m}_{x_t|y_{1:t}} := \frac{1}{n} \sum_{i=1}^n k_{\mathcal{X}}(\cdot, \bar{X}_{t,i}). \tag{4.8}$$

This would be close to (4.7) in the RKHS, as proved by the theoretical analysis in Section 3.4. Then this estimate and pseudo samples are used in the next prediction step at time $t+1$. The analysis in Chapter 3 shows that this procedure can reduce the error of the prediction step.

**Overall algorithm.** We summarize the overall procedure of KMCF in Algorithm 4, where $p_{\text{init}}$ denotes a prior distribution for the initial state $x_1$. For each time $t$, KMCF takes as input an observation $y_t$, and outputs a weight vector $w_t = (w_{t,1}, \ldots, w_{t,n})^T \in \mathbb{R}^n$. Combined with the samples $X_1, \ldots, X_n$ in the state-observation examples $\{(X_i, Y_i)\}_{i=1}^n$, these weights provide an estimator (4.7) of the kernel mean of posterior (4.2).

We first compute kernel matrices $G_X, G_Y$ (Line 4-5), which are used in Algorithm 1 of Kernel Bayes' Rule (Line 15). For $t = 1$, we generate an i.i.d. sample $X_{1,1}, \ldots, X_{1,n}$ from the initial distribution $p_{\text{init}}$ (Line 8), which provides an estimator of the prior

[1] We found in our preliminary experiments that normalization of weights is beneficial to the filtering performance. Such a normalization procedure may be justified with a theoretical analysis by Kanagawa and Fukumizu (2014), which shows the following holds under some mild conditions: Let $\hat{m}_P = \sum_{i=1}^n w_i k_{\mathcal{X}}(\cdot, X_i)$ be an estimator of a kernel mean $m_P$. Then the sum of weights $\sum_{i=1}^n w_i$ converges to 1, as the accuracy of the estimate $\hat{m}_P$ increases, i.e., $\|\hat{m}_P - m_P\|_{\mathcal{H}_{\mathcal{X}}} \to 0$.

**Algorithm 4** Kernel Monte Carlo Filter

1: **Input:** $y_1, \ldots, y_T \in \mathcal{Y}$.
2: **Output:** $w_1, \ldots, w_T \in \mathbb{R}^n$.
3: **Requirement:** $k_{\mathcal{X}}$, $k_{\mathcal{Y}}$, $\varepsilon, \delta$, $\{(X_i, Y_i)\}_{i=1}^n$, $p(x_t|x_{t-1})$, $p_{\mathrm{init}}$.

---

4: $G_X \leftarrow (k_{\mathcal{X}}(X_i, X_j)) \in \mathbb{R}^{n \times n}$.
5: $G_Y \leftarrow (k_{\mathcal{Y}}(Y_i, Y_j)) \in \mathbb{R}^{n \times n}$.
6: **for** $t = 1$ to $T$ **do**
7:  **if** $t = 1$ **then**
8:  Sampling: $X_{1,1}, \ldots, X_{1,n} \sim p_{\mathrm{init}}$ i.i.d.
9:  **else**
10:  $\bar{X}_{t-1,1}, \ldots, \bar{X}_{t-1,n} \leftarrow$ Algorithm 2$(w_{t-1}, \{X_i\}_{i=1}^n)$.
11:  Sampling: $X_{t,i} \sim p(x_t|x_{t-1} = \bar{X}_{t-1,i})$ $(i = 1, \ldots, n)$.
12:  **end if**
13:  $\mathbf{m}_\pi \leftarrow (\frac{1}{n}\sum_{i=1}^n k_{\mathcal{X}}(X_q, X_{t,i}))_{q=1}^n \in \mathbb{R}^n$.
14:  $\mathbf{k}_Y \leftarrow (k_{\mathcal{Y}}(Y_q, y_t))_{q=1}^n \in \mathbb{R}^n$.
15:  $w_t \leftarrow$ Algorithm 1$(\mathbf{k}_Y, \mathbf{m}_\pi, G_X, G_Y, \varepsilon, \delta)$.
16:  $w_t \leftarrow w_t / \sum_{i=1}^n w_{t,i}$.
17: **end for**

corresponding to (4.5). Line 10 is the resampling step at time $t-1$, and Line 11 is the prediction step at time $t$. Line 13-16 corresponds to the correction step.

### 4.2.3 Discussion

The estimation accuracy of KMCF can depend on several factors in practice, and here we discuss them.

**Training samples.** We first note that training samples $\{(X_i, Y_i)\}_{i=1}^n$ should provide the information concerning the observation model $p(y_t|x_t)$. For example, $\{(X_i, Y_i)\}_{i=1}^n$ may be an i.i.d. sample from a joint distribution $p(x, y)$ on $\mathcal{X} \times \mathcal{Y}$, which decomposes as $p(x, y) = p(y|x)p(x)$. Here $p(y|x)$ is the observation model and $p(x)$ is some distribution on $\mathcal{X}$. The support of $p(x)$ should cover the region where states $x_1, \ldots, x_T$ may pass in the test phase, as discussed in Section 4.2.2. For example, this is satisfied when the state space $\mathcal{X}$ is compact, and the support of $p(x)$ is the entire $\mathcal{X}$.

Note that training samples $\{(X_i, Y_i)\}_{i=1}^n$ can also be non-i.i.d in practice. For example, we may deterministically select $X_1, \ldots, X_n$ so that they cover the region of interest. In location estimation problems in robotics, for instance, we may collect location-sensor examples $\{(X_i, Y_i)\}_{i=1}^n$ so that locations $X_1, \ldots, X_n$ cover the region where location estimation is to be conducted (Quigley et al., 2010).

**Hyper-parameters.** As in other kernel methods in general, the performance of KMCF depends on the choice of its hyper-parameters, which are the kernels $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ (or parameters in the kernels, e.g., the bandwidth of the Gaussian kernel) and the regularization constants $\delta, \varepsilon > 0$. We need to define these hyper-parameters based on the joint sample $\{(X_i, Y_i)\}_{i=1}^n$, before running the algorithm on the test data $y_1, \ldots, y_T$. This can be done by cross validation. Suppose that $\{(X_i, Y_i)\}_{i=1}^n$ is given as a sequence from the state-space model. We can then apply two-fold cross validation, by dividing the sequence into two subsequences. If $\{(X_i, Y_i)\}_{i=1}^n$ is not a sequence, we can rely on the cross validation procedure for Kernel Bayes' Rule (see Section 4.2 of Fukumizu et al. (2013)).

**Time complexity.** For each time $t$, the naive implementation of Algorithm 4 requires a time complexity of $O(n^3)$ for the size $n$ of the joint sample $\{(X_i, Y_i)\}_{i=1}^n$. This comes from Algorithm 1 in Line 15 (Kernel Bayes' Rule) and Algorithm 2 in Line 10 (resampling). The complexity $O(n^3)$ of Algorithm 1 is due to the matrix inversions. Note that one of the inversions $(G_X + n\varepsilon I_n)^{-1}$ can be computed before the test phase, as it does not involve the test data. Algorithm 2 also has complexity of $O(n^3)$. In Section 3.3, we explained how this cost can be reduced to $O(n^2\ell)$ by generating only $\ell < n$ samples by resampling.

**Speeding up methods.** In Section 4.3, we describe two methods for reducing the computational costs of KMCF, both of which only need to be applied prior to the test phase. (i) Low rank approximation of kernel matrices $G_X, G_Y$, which reduces the complexity to $O(nr^2)$, where $r$ the rank of low rank matrices: Low rank approximation works well in practice, since eigenvalues of a kernel matrix often decay very rapidly. Indeed this has been theoretically shown for some cases; see Widom (1963, 1964) and discussions in Bach and Jordan (2002). (ii) A data reduction method based on Kernel Herding, which efficiently selects joint subsamples from the training set $\{(X_i, Y_i)\}_{i=1}^n$: Algorithm 4 is then applied based only on those subsamples. The resulting complexity is thus $O(r^3)$, where $r$ is the number of subsamples. This method is motivated by the fast convergence rate of Kernel Herding (Chen et al., 2010).

Both methods require the number $r$ to be chosen, which is either the rank for low rank approximation, or the number of subsamples in data reduction. This determines the tradeoff between the accuracy and computational time. In practice, there are two ways of selecting the number $r$. (a) By regarding $r$ as a hyper parameter of KMCF, we can select it by cross validation. (b) We can choose $r$ by comparing the resulting approximation error; such error is measured in a matrix norm for low rank approximation, and in an RKHS norm for the subsampling method. For details, see Section 4.3.

**Transfer leaning setting.** We assumed that the observation model in the test phase is the same as for the training samples. However, this might not hold in some situations. For example, in the vision-based localization problem, the illumination conditions for the test and training phases might be different (e.g., the test is done at night, while the training samples are collected in the morning). Without taking into account such a significant change in the observation model, KMCF would not perform well in practice.

This problem could be addressed by exploiting the framework of *transfer learning* (Pan and Yang, 2010). This framework aims at situations where the probability distribution that generates test data is different from that of training samples. The main assumption is that there exist a small number of examples from the test distribution. Transfer learning then provides a way of combining such test examples and abundant training samples, thereby improving the test performance. The application of transfer learning in our setting remains a topic for future research.

### 4.2.4 Estimation of posterior statistics

By Algorithm 4, we obtain the estimates of the kernel means of posteriors (4.2) as

$$\hat{m}_{x_t|y_{1:t}} = \sum_{i=1}^{n} w_{t,i} k_{\mathcal{X}}(\cdot, X_i) \quad (t = 1, \ldots, T). \tag{4.9}$$

These contain the information on the posteriors $p(x_t|y_{1:t})$ (see Section 2.2). We now show how to estimate statistics of the posteriors using these estimates (4.9). For ease of presentation, we consider the case $\mathcal{X} = \mathbb{R}^d$. A theoretical background to justify these operations is provided in Chapter 5.

**Mean and covariance.** Consider the posterior mean $\int x_t p(x_t|y_{1:t}) dx_t \in \mathbb{R}^d$ and the posterior (uncentered) covariance $\int x_t x_t^T p(x_t|y_{1:t}) dx_t \in \mathbb{R}^{d \times d}$. These quantities can be estimated as

$$\sum_{i=1}^{n} w_{t,i} X_i \ \text{(mean)}. \qquad \sum_{i=1}^{n} w_{t,i} X_i X_i^T \ \text{(covariance)}.$$

**Probability mass.** Let $A \subset \mathcal{X}$ be a measurable set with smooth boundary. Define the indicator function $I_A(x)$ by $I_A(x) = 1$ for $x \in A$ and $I_A(x) = 0$ otherwise. Consider the probability mass $\int I_A(x) p(x_t|y_{1:t}) dx_t$. This can be estimated as $\sum_{i=1}^{n} w_{t,i} I_A(X_i)$.

**Density.** Suppose $p(x_t|y_{1:t})$ has a density function. Let $J(x)$ be a smoothing kernel satisfying $\int J(x)dx = 1$ and $J(x) \geq 0$. Let $h > 0$ and define $J_h(x) := \frac{1}{h^d} J\left(\frac{x}{h}\right)$. Then the density of $p(x_t|y_{1:t})$ can be estimated as

$$\hat{p}(x_t|y_{1:t}) = \sum_{i=1}^{n} w_{t,i} J_h(x_t - X_i), \tag{4.10}$$

with an appropriate choice of $h$.

**Mode.** The mode may be obtained by finding a point that maximizes (4.10). However, this requires a careful choice of $h$. Instead, we may use $X_{i_{\max}}$ with $i_{\max} := \arg\max_i w_{t,i}$ as a mode estimate: this is the point in $\{X_1, \ldots, X_n\}$ that is associated with the maximum weight in $w_{t,1}, \ldots, w_{t,n}$. This point can be interpreted as the point that maximizes (4.10) in the limit of $h \to 0$.

**Other methods.** Other ways of using (4.9) include the pre-image computation and fitting of Gaussian mixtures. See, e.g., Song et al. (2009); Fukumizu et al. (2013); McCalman et al. (2013).

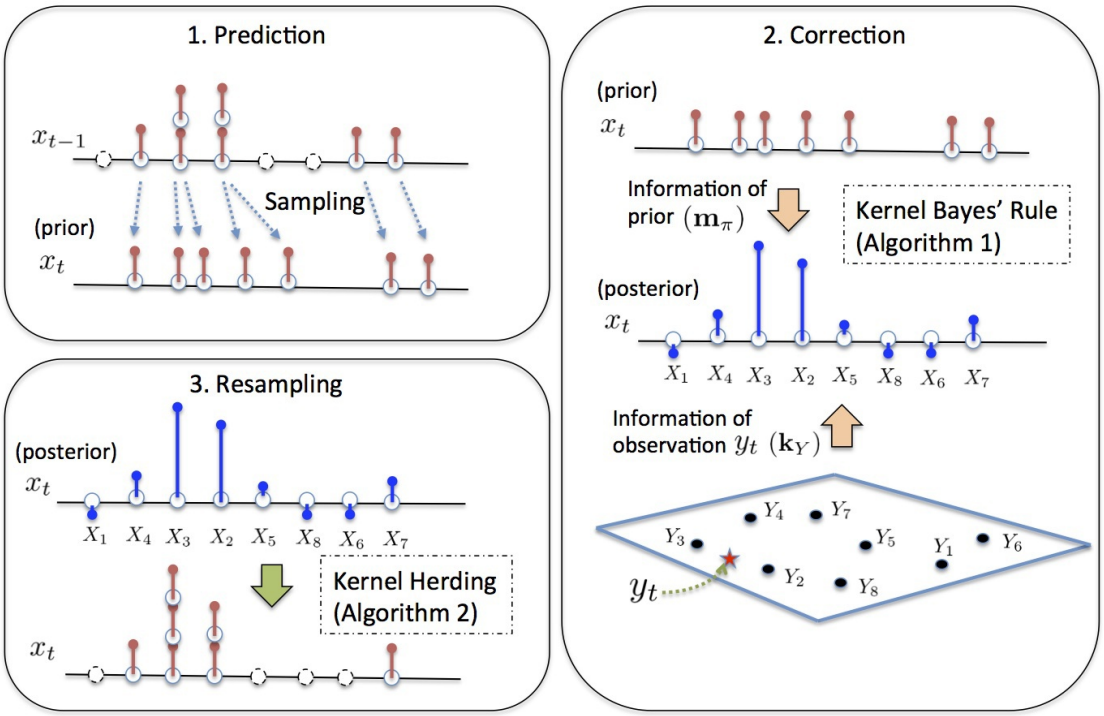



Figure 4.2: One iteration of KMCF. Here $X_1, \ldots, X_8$ and $Y_1, \ldots, Y_8$ denote states and observations, respectively, in the state-observation examples $\{(X_i, Y_i)\}_{i=1}^n$ (suppose $n = 8$). **1. Prediction step:** The kernel mean of the prior (4.6) is estimated by sampling with the transition model $p(x_t|x_{t-1})$. **2. Correction step:**. The kernel mean of the posterior (4.2) is estimated by applying Kernel Bayes' Rule (Algorithm 1). The estimation makes use of the information of the prior (expressed as $\mathbf{m}_\pi := (\hat{m}_{x_t|y_{1:t-1}}(X_i)) \in \mathbb{R}^8$) as well as that of a new observation $y_t$ (expressed as $\mathbf{k}_Y := (k_\mathcal{Y}(y_t, Y_i)) \in \mathbb{R}^8$). The resulting estimate (4.7) is expressed as a weighted sample $\{(w_{t,i}, X_i)\}_{i=1}^n$. Note that the weights may be negative. **3. Resampling step:** Samples associated with small weights are eliminated, and those with large weights are replicated by applying Kernel Herding (Algorithm 2). The resulting samples provide an empirical kernel mean (4.8), which will be used in the next iteration.

## 4.3 Acceleration methods

We have seen in Section 4.2.3 that the time complexity of KMCF in one time step is $O(n^3)$, where $n$ is the number of the state-observation examples $\{(X_i, Y_i)\}_{i=1}^n$. This can be costly if one wishes to use KMCF in real-time applications with a large number of samples. Here we show two methods for reducing the costs: one based on low rank approximation of kernel matrices, and one based on Kernel Herding. Note that Kernel Herding is also used in the resampling step. The purpose here is different, however: we make use of Kernel Herding for finding a reduced representation of the data $\{(X_i, Y_i)\}_{i=1}^n$.

### 4.3.1 Low rank approximation of kernel matrices

Our goal is to reduce the costs of Algorithm 1 of Kernel Bayes' Rule. Algorithm 1 involves two matrix inversions: $(G_X + n\varepsilon I_n)^{-1}$ in Line 3 and $((\Lambda G_Y)^2 + \delta I_n)^{-1}$ in Line 4. Note that $(G_X + n\varepsilon I_n)^{-1}$ does not involve the test data, so can be computed before the test phase. On the other hand, $((\Lambda G_Y)^2 + \delta I_n)^{-1}$ depends on matrix $\Lambda$. This matrix involves the vector $\mathbf{m}_\pi$, which essentially represents the prior of the current state (see Line 13 of Algorithm 4). Therefore $((\Lambda G_Y)^2 + \delta I_n)^{-1}$ needs to be computed for each iteration in the test phase. This has complexity of $O(n^3)$. Note that even if $(G_X + n\varepsilon I_n)^{-1}$ can be computed in the training phase, the multiplication $(G_X + n\varepsilon I_n)^{-1}\mathbf{m}_\pi$ in Line 3 requires $O(n^2)$. Thus it can also be costly. Here we consider methods to reduce both costs in Line 3 and 4.

Suppose that there exist low rank matrices $U, V \in \mathbb{R}^{n\times r}$, where $r < n$, that approximate the kernel matrices: $G_X \approx UU^T$, $G_Y \approx VV^T$. Such low rank matrices can be obtained by, for example, incomplete Cholesky decomposition with time complexity $O(nr^2)$ (Fine and Scheinberg, 2001; Bach and Jordan, 2002). Note that the computation of these matrices are only required once before the test phase. Therefore their time complexities are not the problem here.

**Derivation.** First, we approximate $(G_X + n\varepsilon I_n)^{-1}\mathbf{m}_\pi$ in Line 3 using $G_X \approx UU^T$. By the Woodbury identity, we have

$$\begin{aligned}(G_X + n\varepsilon I_n)^{-1}\mathbf{m}_\pi &\approx (UU^T + n\varepsilon I_n)^{-1}\mathbf{m}_\pi \\ &= \frac{1}{n\varepsilon}(I_n - U(n\varepsilon I_r + U^TU)^{-1}U^T)\mathbf{m}_\pi,\end{aligned}$$

where $I_r \in \mathbb{R}^{r\times r}$ denotes the identity. Note that $(n\varepsilon I_r + U^TU)^{-1}$ does not involve the test data, so can be computed in the training phase. Thus the above approximation of $\mu$ can be computed with complexity $O(nr^2)$.

**Algorithm 5** Low Rank Approximation of Kernel Bayes' Rule

1: **Input:** $\mathbf{k}_Y, \mathbf{m}_\pi \in \mathbb{R}^n$, $U, V \in \mathbb{R}^{n\times r}$, $\varepsilon, \delta > 0$.
2: **Output:** $w := (w_1, \ldots, w_n)^T \in \mathbb{R}^n$.

3: $\Lambda \leftarrow \mathrm{diag}(\frac{1}{n\varepsilon}(I_n - U(n\varepsilon I_r + U^T U)^{-1}U^T)\mathbf{m}) \in \mathbb{R}^{n\times n}$.
4: $B \leftarrow \Lambda V \in \mathbb{R}^{n\times r}, \quad C \leftarrow V^T \Lambda V \in \mathbb{R}^{r\times r}, \quad D \leftarrow V^T \in \mathbb{R}^{r\times n}$.
5: $w \leftarrow \frac{1}{\delta}\Lambda V V^T (I_n - B(\delta C^{-1} + DB)^{-1}D)\Lambda \mathbf{k}_Y \in \mathbb{R}^n$.

Next, we approximate $w = \Lambda G_Y((\Lambda G_Y)^2 + \delta I)^{-1}\Lambda \mathbf{k}_Y$ in Line 4 using $G_Y \approx VV^T$. Define $B = \Lambda V \in \mathbb{R}^{n\times r}$, $C = V^T \Lambda V \in \mathbb{R}^{r\times r}$, and $D = V^T \in \mathbb{R}^{r\times n}$. Then $(\Lambda G_Y)^2 \approx (\Lambda V V^T)^2 = BCD$. By the Woodbury identity, we obtain

$$\begin{aligned}(\delta I_n + (\Lambda G_Y)^2)^{-1} &\approx (\delta I_n + BCD)^{-1} \\ &= \frac{1}{\delta}(I_n - B(\delta C^{-1} + DB)^{-1}D).\end{aligned}$$

Thus $w$ can be approximated as

$$\begin{aligned}w &= \Lambda G_Y((\Lambda G_Y)^2 + \delta I)^{-1}\Lambda \mathbf{k}_Y \\ &\approx \frac{1}{\delta}\Lambda V V^T (I_n - B(\delta C^{-1} + DB)^{-1}D)\Lambda \mathbf{k}_Y.\end{aligned}$$

The computation of this approximation requires $O(nr^2 + r^3) = O(nr^2)$. Thus in total, the complexity of Algorithm 1 can be reduced to $O(nr^2)$. We summarize the above approximations in Algorithm 5.

**How to select the rank.** As discussed in Section 4.2.3, one way of selecting the rank $r$ is to use cross validation, by regarding $r$ as a hyper parameter of KMCF. Another way is to measure the approximation errors $\|G_X - UU^T\|$ and $\|G_Y - VV^T\|$ with some matrix norm, such as the Frobenius norm. Indeed, we can compute the smallest rank $r$ such that these errors are below a prespecified threshold, and this can be done efficiently with time complexity $O(nr^2)$ (Bach and Jordan, 2002).

### 4.3.2 Data reduction with Kernel Herding

Here we describe an approach to reduce the size of the representation of the state-observation examples $\{(X_i, Y_i)\}_{i=1}^n$ in an efficient way. By "efficient", we mean that the information contained in $\{(X_i, Y_i)\}_{i=1}^n$ will be preserved even after the reduction. Recall that $\{(X_i, Y_i)\}_{i=1}^n$ contains the information of the observation model $p(y_t|x_t)$ (recall also that $p(y_t|x_t)$ is assumed time-invariant; see Section 4.2.1). This infor-

mation is only used in Algorithm 1 of Kernel Bayes' Rule (Line 15, Algorithm 4). Therefore it suffices to consider how Kernel Bayes' Rule accesses the information contained in the joint sample $\{(X_i, Y_i)\}_{i=1}^n$.

**Representation of the joint sample.** To this end, we need to show how the joint sample $\{(X_i, Y_i)\}_{i=1}^n$ can be represented with a kernel mean embedding. Recall that $(k_{\mathcal{X}}, \mathcal{H}_{\mathcal{X}})$ and $(k_{\mathcal{Y}}, \mathcal{H}_{\mathcal{Y}})$ are kernels and the associated RKHSs on the state space $\mathcal{X}$ and the observation space $\mathcal{Y}$, respectively. Let $\mathcal{X} \times \mathcal{Y}$ be the product space of $\mathcal{X}$ and $\mathcal{Y}$. Then we can define a kernel $k_{\mathcal{X}\times\mathcal{Y}}$ on $\mathcal{X} \times \mathcal{Y}$ as the product of $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$: $k_{\mathcal{X}\times\mathcal{Y}}((x, y), (x', y')) = k_{\mathcal{X}}(x, x')k_{\mathcal{Y}}(y, y')$ for all $(x, y), (x', y') \in \mathcal{X} \times \mathcal{Y}$. This product kernel $k_{\mathcal{X}\times\mathcal{Y}}$ defines an RKHS of $\mathcal{X} \times \mathcal{Y}$: let $\mathcal{H}_{\mathcal{X}\times\mathcal{Y}}$ denote this RKHS. As in Section 2, we can use $k_{\mathcal{X}\times\mathcal{Y}}$ and $\mathcal{H}_{\mathcal{X}\times\mathcal{Y}}$ for a kernel mean embedding. In particular, the empirical distribution $\frac{1}{n}\sum_{i=1}^n \delta_{(X_i,Y_i)}$ of the joint sample $\{(X_i, Y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ can be represented as an empirical kernel mean in $\mathcal{H}_{\mathcal{X}\times\mathcal{Y}}$:

$$\hat{m}_{XY} := \frac{1}{n}\sum_{i=1}^n k_{\mathcal{X}\times\mathcal{Y}}((\cdot, \cdot), (X_i, Y_i)) \in \mathcal{H}_{\mathcal{X}\times\mathcal{Y}}. \tag{4.11}$$

This is the representation of the joint sample $\{(X_i, Y_i)\}_{i=1}^n$.

The information of $\{(X_i, Y_i)\}_{i=1}^n$ is provided for Kernel Bayes' Rule essentially through this form (4.11) (Fukumizu et al., 2011, 2013). Recall that (4.11) is a point in the RKHS $\mathcal{H}_{\mathcal{X}\times\mathcal{Y}}$. Any point close to (4.11) in $\mathcal{H}_{\mathcal{X}\times\mathcal{Y}}$ would also contain information close to that contained in (4.11). Therefore, we propose to find a subset $\{(\bar{X}_1, \bar{Y}_1), \ldots (\bar{X}_r, \bar{X}_r)\} \subset \{(X_i, Y_i)\}_{i=1}^n$, where $r < n$, such that its representation in $\mathcal{H}_{\mathcal{X}\times\mathcal{Y}}$

$$\bar{m}_{XY} := \frac{1}{r}\sum_{i=1}^r k_{\mathcal{X}\times\mathcal{Y}}((\cdot, \cdot), (\bar{X}_i, \bar{Y}_i)) \in \mathcal{H}_{\mathcal{X}\times\mathcal{Y}} \tag{4.12}$$

is close to (4.11). Namely, we wish to find subsamples such that $\|\bar{m}_{XY} - \hat{m}_{XY}\|_{\mathcal{H}_{\mathcal{X}\times\mathcal{Y}}}$ is small. If the error $\|\bar{m}_{XY} - \hat{m}_{XY}\|_{\mathcal{H}_{\mathcal{X}\times\mathcal{Y}}}$ is small enough, (4.12) would provide information close to that given by (4.11) for Kernel Bayes' Rule. Thus Kernel Bayes' Rule based on such subsamples $\{(\bar{X}_i, \bar{Y}_i)\}_{i=1}^r$ would not perform much worse than the one based on the entire set of samples $\{(X_i, Y_i)\}_{i=1}^n$.

**Subsampling method.** To find such subsamples, we make use of Kernel Herding in Section 2.7. Namely, we apply the update equations (2.16) (2.17) to approximate (4.11), with kernel $k_{\mathcal{X}\times\mathcal{Y}}$ and RKHS $\mathcal{H}_{\mathcal{X}\times\mathcal{Y}}$. We greedily find subsamples $\bar{\mathbf{D}}_r :=$

$\{(\bar{X}_1, \bar{Y}_1), \ldots, (\bar{X}_r, \bar{Y}_r)\}$ as

$$
\begin{aligned}
(\bar{X}_r, \bar{Y}_r) &= \arg \max_{(x,y) \in \mathbf{D}/\bar{\mathbf{D}}_{r-1}} \frac{1}{n} \sum_{i=1}^{n} k_{\mathcal{X}\times\mathcal{Y}} \left((x,y),(X_i,Y_i)\right) - \frac{1}{r} \sum_{j=1}^{r-1} k_{\mathcal{X}\times\mathcal{Y}} \left((x,r),(\bar{X}_i,\bar{Y}_i)\right) \\
&= \arg \max_{(x,y) \in \mathbf{D}/\bar{\mathbf{D}}_{r-1}} \frac{1}{n} \sum_{i=1}^{n} k_{\mathcal{X}}(x,X_i) k_{\mathcal{Y}}(y,Y_i) - \frac{1}{r} \sum_{j=1}^{r-1} k_{\mathcal{X}}(x,\bar{X}_j) k_{\mathcal{Y}}(y,\bar{Y}_j).
\end{aligned}
$$

The resulting algorithm is shown in Algorithm 6. The time complexity is $O(n^2 r)$ for selecting $r$ subsamples. We propose to use this algorithm before going to the test phase. Once we obtain the subsamples $\{(\bar{X}_i, \bar{Y}_i)\}_{i=1}^{r}$, we can apply Algorithm 4 with these samples instead of the entire set of samples $\{(X_i, Y_i)\}_{i=1}^{n}$. The time complexity of Algorithm 4 for each iteration is then reduced to $O(r^3)$.

**Discussion.** Recall that Kernel Herding generates samples such that they approximate a given kernel mean (see Section 2.7). Under certain assumptions, the error of this approximation is of $O(r^{-1})$ with $r$ samples, which is faster than that of i.i.d. samples $O(r^{-1/2})$. However, these assumptions only hold for finite dimensional RKHSs. Gaussian kernels, which we often use in practice, define infinite dimensional RKHSs. Therefore the fast rate is not guaranteed if we use Gaussian kernels. Nevertheless, we can use Algorithm 6 as a heuristic for data reduction.

**How to select the number of subsamples.** The number $r$ of subsamples determine the tradeoff between the accuracy and computational time of KMCF. It may be selected by cross validation, or by measuring the approximation error $\|\bar{m}_{XY} - \hat{m}_{XY}\|_{\mathcal{H}_{\mathcal{X}\times\mathcal{Y}}}$, as for the case of selecting the rank of low rank approximation in Appendix 4.3.1.

**Algorithm 6** Subsampling with Kernel Herding

1: **Input:** (i) $\mathbf{D} := \{(X_i, Y_i)\}_{i=1}^n$. (ii) size of subsamples $r$.
2: **Output:** subsamples $\bar{\mathbf{D}}_r := \{(\bar{X}_1, \bar{Y}_1), \ldots, (\bar{X}_r, \bar{Y}_r)\} \subset \mathbf{D}$.

3: Select $(\bar{X}_1, \bar{Y}_1)$ as follows and let $\bar{\mathbf{D}}_1 := \{(\bar{X}_1, \bar{Y}_1)\}$:

$$(\bar{X}_1, \bar{Y}_1) = \underset{(x,y)\in\mathbf{D}}{\arg\max} \frac{1}{n} \sum_{i=1}^{n} k_{\mathcal{X}}(x, X_i) k_{\mathcal{Y}}(y, Y_i)$$

4: **for** $N = 2$ to $r$ **do**
5: Select $(\bar{X}_N, \bar{Y}_N)$ as follows and let $\bar{\mathbf{D}}_N := \bar{\mathbf{D}}_{N-1} \cup \{(\bar{X}_N, \bar{Y}_N)\}$:

$$(\bar{X}_N, \bar{Y}_N) = \underset{(x,y)\in\mathbf{D}/\bar{\mathbf{D}}_{N-1}}{\arg\max} \frac{1}{n} \sum_{i=1}^{n} k_{\mathcal{X}}(x, X_i) k_{\mathcal{Y}}(u, Y_i) - \frac{1}{N} \sum_{j=1}^{N-1} k_{\mathcal{X}}(x, \bar{X}_j) k_{\mathcal{Y}}(y, \bar{Y}_j)$$

6: **end for**

## 4.4 Theoretical analysis

Here we show the consistency of the overall filtering procedure of KMCF. This is based on Corollary 2 in Section 3.4, which shows the consistency of the resampling step followed by the prediction step, and on Theorem 5 of Fukumizu et al. (2013), which guarantees the consistency of Kernel Bayes' Rule in the correction step. Thus we consider three steps in the following order: (i) resampling; (ii) prediction; (iii) correction. More specifically, we show consistency of the estimator (4.7) of the posterior kernel mean at time $t$, given that the one at time $t-1$ is consistent.

To state our assumptions, we will need the following functions $\theta_{\text{pos}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, $\theta_{\text{obs}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, and $\theta_{\text{tra}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$:

$$\theta_{\text{pos}}(y, \tilde{y}) := \int\int k_{\mathcal{X}}(x_t, \tilde{x}_t) dp(x_t | y_{1:t-1}, y_t = y) dp(\tilde{x}_t | y_{1:t-1}, y_t = \tilde{y}), \quad (4.13)$$

$$\theta_{\text{obs}}(x, \tilde{x}) := \int\int k_{\mathcal{Y}}(y_t, \tilde{y}_t) dp(y_t | x_t = x) dp(\tilde{y}_t | x_t = \tilde{x}), \quad (4.14)$$

$$\theta_{\text{tra}}(x, \tilde{x}) := \int\int k_{\mathcal{X}}(x_t, \tilde{x}_t) dp(x_t | x_{t-1} = x) dp(\tilde{x}_t | x_{t-1} = \tilde{x}). \quad (4.15)$$

These functions contain the information concerning the distributions involved. In (4.13), the distribution $p(x_t | y_{1:t-1}, y_t = y)$ denotes the posterior of the state at time $t$, given that the observation at time $t$ is $y_t = y$. Similarly $p(\tilde{x}_t | y_{1:t-1}, y_t = \tilde{y})$ is the posterior at time $t$, given that the observation is $y_t = \tilde{y}_t$. In (4.14), the distributions

$p(y_t|x_t = x)$ and $p(\tilde{y}_t|x_t = \tilde{x})$ denote the observation model when the state is $x_t = x$ or $x_t = \tilde{x}$, respectively. In (4.15), the distributions $p(x_t|x_{t-1} = x)$ and $p(\tilde{x}_t|x_{t-1} = \tilde{x})$ denote the transition model with the previous state given by $x_{t-1} = x$ or $x_{t-1} = \tilde{x}$, respectively.

Below denote by $\mathcal{F} \otimes \mathcal{G}$ the tensor product space of two RKHSs $\mathcal{F}$ and $\mathcal{G}$.

**Corollary 3.** *Let $(X_1, Y_1), \ldots, (X_n, Y_n)$ be an i.i.d. sample with a joint density $p(x, y) := p(y|x)q(x)$, where $p(y|x)$ is the observation model. Assume that the posterior $p(x_t|y_{1:t})$ has a density $p$, and that $\sup_{x \in \mathcal{X}} p(x)/q(x) < \infty$. Assume that the functions defined by (4.13), (4.14) and (4.15) satisfy $\theta_{\mathrm{pos}} \in \mathcal{H}_{\mathcal{Y}} \otimes \mathcal{H}_{\mathcal{Y}}$, $\theta_{\mathrm{obs}} \in \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}$ and $\theta_{\mathrm{tra}} \in \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}$, respectively. Suppose that $\|\hat{m}_{x_{t-1}|y_{1:t-1}} - m_{x_{t-1}|y_{1:t-1}}\|_{\mathcal{H}_{\mathcal{X}}} \to 0$ as $n \to \infty$ in probability. Then for any sufficiently slow decay of regularization constants $\varepsilon_n$ and $\delta_n$ of Algorithm 1, we have*

$$\|\hat{m}_{x_t|y_{1:t}} - m_{x_t|y_{1:t}}\|_{\mathcal{H}_{\mathcal{X}}} \to 0 \quad (n \to \infty)$$

*in probability.*

Corollary 3 follows from Theorem 5 of Fukumizu et al. (2013) and Corollary 2. The assumptions $\theta_{\mathrm{pos}} \in \mathcal{H}_{\mathcal{Y}} \otimes \mathcal{H}_{\mathcal{Y}}$ and $\theta_{\mathrm{obs}} \in \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}$ are due to Theorem 5 of Fukumizu et al. (2013) for the correction step, while the assumption $\theta_{\mathrm{tra}} \in \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{X}}$ is due to Theorem 1 for the prediction step, from which Corollary 2 follows. As we discussed in footnote 4 of Section 3.1, these essentially assume that the functions $\theta_{\mathrm{pos}}$, $\theta_{\mathrm{obs}}$ and $\theta_{\mathrm{tra}}$ are smooth. Theorem 5 of Fukumizu et al. (2013) also requires that the regularization constants $\varepsilon_n, \delta_n$ of Kernel Bayes' Rule should decay sufficiently slowly, as the sample size goes to infinity ($\varepsilon_n, \delta_n \to 0$ as $n \to \infty$). For details, see Sections 5.2 and 6.2 in Fukumizu et al. (2013).

It would be more interesting to investigate the convergence rates of the overall procedure. However, this requires a refined theoretical analysis of Kernel Bayes' Rule, which is beyond the scope of this chapter. This is because currently there is no theoretical result on convergence rates of Kernel Bayes' Rule as an estimator of a posterior kernel mean (existing convergence results are for the expectation of function values; see Theorems 6 and 7 in Fukumizu et al. (2013)). This remains a topic for future research.

## 4.5 Experiments

This section is devoted to experiments. In Section 4.5.1, the proposed KMCF (Algorithm 4) is applied to synthetic state-space models. Comparisons are made with existing methods applicable to the setting of the paper. In Section 4.5.2, we apply KMCF to the real problem of vision-based robot localization.

In the following, $\mathbb{N}(\mu, \sigma^2)$ denotes the Gaussian distribution with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 > 0$.

### 4.5.1 Filtering with synthetic state-space models

Here we apply KMCF to synthetic state-space models. Comparisons were made with the following methods:

**kNN-PF (Vlassis et al., 2002)** This method uses $k$-NN-based conditional density estimation Stone (1977) for learning the observation model. First, it estimates the conditional density of the inverse direction $p(x|y)$ from the training sample $\{(X_i, Y_i)\}$. The learned conditional density is then used as an alternative for the likelihood $p(y_t|x_t)$; this is a heuristic to deal with high-dimensional $y_t$. Then it applies Particle Filter (PF), based on the approximated observation model and the given transition model $p(x_t|x_{t-1})$.

**GP-PF (Ferris et al., 2006)** This method learns $p(y_t|x_t)$ from $\{(X_i, Y_i)\}$ with Gaussian Process (GP) regression. Then Particle Filter is applied based on the learned observation model and the transition model. We used the open-source code[2] for GP-regression in this experiment, so comparison in computational time is omitted for this method.

**KBR filter (Fukumizu et al., 2011, 2013)** This method is also based on kernel mean embeddings, as is KMCF. It applies Kernel Bayes' Rule (KBR) in posterior estimation using the joint sample $\{(X_i, Y_i)\}$. This method assumes that there also exist training samples for the transition model. Thus in the following experiments, we additionally drew training samples for the transition model. It was shown (Fukumizu et al., 2011, 2013) that this method outperforms Extended and Unscented Kalman Filters, when a state-space model has strong nonlinearity (in that experiment, these Kalman filters were given the full-knowledge of a state-space model). We use this method as a baseline.

We used state-space models defined in Table 4.2, where SSM stands for State Space Model. In Table 4.2, $u_t$ denotes a control input at time $t$; $v_t$ and $w_t$ denote independent Gaussian noise: $v_t, w_t \sim \mathbb{N}(0, 1)$; $W_t$ denotes 10 dimensional Gaussian noise: $W_t \sim \mathbb{N}(0, I_{10})$. We generated each control $u_t$ randomly from the Gaussian distribution $\mathbb{N}(0, 1)$.

The state and observation spaces for SSMs {1a, 1b, 2a, 2b, 4a, 4b} are defined as $\mathcal{X} = \mathcal{Y} = \mathbb{R}$; for SSMs {3a, 3b}, $\mathcal{X} = \mathbb{R}, \mathcal{Y} = \mathbb{R}^{10}$. The models in SSMs {1a, 2a, 3a, 4a} and SSMs {1b, 2b, 3b, 4b} with the same number (e.g., 1a and 1b) are almost the same; the difference is whether $u_t$ exists in the transition model. Prior distributions for the initial state $x_1$ for SSMs {1a, 1b, 2a, 2b, 3a, 3b} are defined as

[2] http://www.gaussianprocess.org/gpml/code/matlab/doc/

$p_{\text{init}} = \mathbb{N}(0, 1/(1-0.9^2))$, and those for {4a, 4b} are defined as a uniform distribution on $[-3, 3]$.

Table 4.2: State-space models (SSM) for synthetic experiments

| SSM | transition model | observation model |
|---|---|---|
| 1a | $x_t = 0.9x_{t-1} + v_t$ | $y_t = x_t + w_t$ |
| 1b | $x_t = 0.9x_{t-1} + \frac{1}{\sqrt{2}}(u_t + v_t)$ | $y_t = x_t + w_t$ |
| 2a | $x_t = 0.9x_{t-1} + v_t$ | $y_t = 0.5\exp(x_t/2)w_t$ |
| 2b | $x_t = 0.9x_{t-1} + \frac{1}{\sqrt{2}}(u_t + v_t)$ | $y_t = 0.5\exp(x_t/2)w_t$ |
| 3a | $x_t = 0.9x_{t-1} + v_t$ | $y_t = 0.5\exp(x_t/2)W_t$ |
| 3b | $x_t = 0.9x_{t-1} + \frac{1}{\sqrt{2}}(u_t + v_t)$ | $y_t = 0.5\exp(x_t/2)W_t$ |
| 4a | $a_t = x_{t-1} + \sqrt{2}v_t$<br>$x_t = \begin{cases} a_t & (\text{if } \lvert a_t\rvert \le 3) \\ -3 & (\text{otherwise}) \end{cases}$ | $b_t = x_t + w_t$<br>$y_t = \begin{cases} b_t & (\text{if } \lvert b_t\rvert \le 3) \\ b_t - 6b_t/\lvert b_t\rvert & (\text{otherwise}) \end{cases}$ |
| 4b | $a_t = x_{t-1} + u_t + v_t$<br>$x_t = \begin{cases} a_t & (\text{if } \lvert a_t\rvert \le 3) \\ -3 & (\text{otherwise}) \end{cases}$ | $b_t = x_t + w_t$<br>$y_t = \begin{cases} b_t & (\text{if } \lvert b_t\rvert \le 3) \\ b_t - 6b_t/\lvert b_t\rvert & (\text{otherwise}) \end{cases}$ |

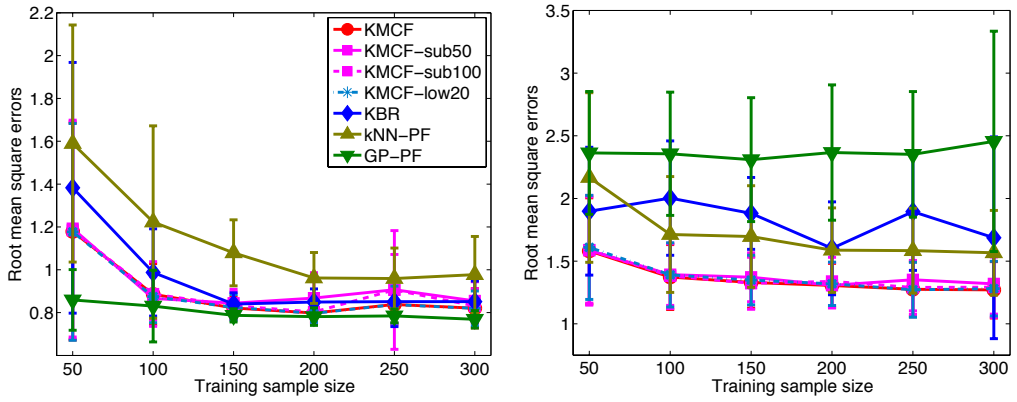
SSM 1a and 1b are linear Gaussian models. SSM 2a and 2b are the so-called stochastic volatility models. Their transition models are the same as those of SSM 1a and 1b. On the other hand, the observation model has strong nonlinearity and the noise $w_t$ is multiplicative. SSM 3a and 3b are almost the same as SSM 2a and 2b. The difference is that the observation $y_t$ is 10 dimensional, as $W_t$ is 10 dimensional Gaussian noise. SSM 4a and 4b are more complex than the other models. Both the transition and observation models have strong nonlinearities: states and observations located around the edges of the interval $[-3, 3]$ may abruptly jump to distant places.

For each model, we generated the training samples $\{(X_i, Y_i)\}_{i=1}^n$ by simulating the model. Test data $\{(x_t, y_t)\}_{t=1}^T$ was also generated by independent simulation (recall that $x_t$ is hidden for each method). The length of the test sequence was set as $T = 100$. We fixed the number of particles in kNN-PF and GP-PF to 5000; in primary experiments, we did not observe any improvements even when more particles were used. For the same reason, we fixed the size of transition examples for KBR filter to 1000. Each method estimated the ground truth states $x_1, \ldots, x_T$ by estimating the posterior means $\int x_t p(x_t|y_{1:t})dx_t$ $(t = 1, \ldots, T)$. The performance was evaluated with RMSE (Root Mean Squared Errors) of the point estimates, defined as $RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^T (\hat{x}_t - x_t)^2}$, where $\hat{x}_t$ is the point estimate.

For KMCF and KBR filter, we used Gaussian kernels for each of $\mathcal{X}$ and $\mathcal{Y}$ (and also for controls in KBR filter). We determined the hyper-parameters of each method by two-fold cross validation, by dividing the training data into two sequences. The hyper-parameters in the GP-regressor for PF-GP were optimized by maximizing the marginal likelihood of the training data. To reduce the costs of the resampling step of KMCF, we used the method discussed in Section 3.3 with $\ell = 50$. We also used the low rank approximation method (Algorithm 5) and the subsampling method (Algorithm 6) in Appendix 4.3 to reduce the computational costs of KMCF. Specifically, we used $r = 10, 20$ (rank of low rank matrices) for Algorithm 5 (described as KMCF-low10 and KMCF-low20 in the results below); $r = 50, 100$ (number of subsamples) for Algorithm 6 (described as KMCF-sub50 and KMCF-sub100). We repeated experiments 20 times for each of different training sample size $n$.

Figure 4.3 shows the results in RMSE for SSMs {1a, 2a, 3a, 4a}, and Figure 4.4 shows those for SSMs {1b, 2b, 3b, 4b}. Figure 4.5 describes the results in computational time for SSM 1a and 1b; the results for the other models are similar, so we omit them. We do not show the results of KMCF-low10 in Figure 4.3 and 4.4, since they were numerically unstable and gave very large RMSEs.

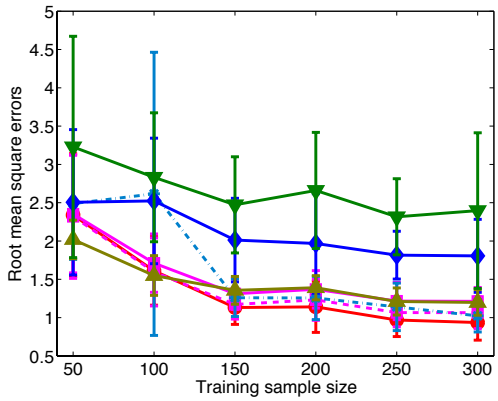
GP-PF performed the best for SSM 1a and 1b. This may be because these models fit the assumption of GP-regression, as their noise are additive Gaussian. For the other models, however, GP-PF performed poorly; the observation models of these models have strong nonlinearities and the noise are not additive Gaussian. For these models, KMCF performed the best or competitively with the other methods. This indicates that KMCF successfully exploits the state-observation examples $\{(X_i, Y_i)\}_{i=1}^n$ in dealing with the complicated observation models. Recall that our focus has been on situations where the relation between states and observations are so complicated that the observation model is not known; the results indicate that KMCF is promising for such situations. On the other hand, KBR filter performed worse than KMCF for the most of the models. KBF filter also uses Kernel Bayes' Rule as KMCF. The difference is that KMCF makes use of the transition models directly by sampling, while KBR filter must learn the transition models from training data for state transitions. This indicates that the incorporation of the knowledge expressed in the transition model is very important for the filtering performance. This can also be seen by comparing Figure 4.3 and Figure 4.4. The performance of the methods other than KBR filter improved for SSMs {1b, 2b, 3b, 4b}, compared to the performance for the corresponding models in SSMs {1a, 2a, 3a, 4a}. Recall that SSMs {1b, 2b, 3b, 4b} include control $u_t$ in their transition models. The information of control input is helpful for filtering in general. Thus the improvements suggest that KMCF, kNN-PF and GP-PF successfully incorporate the information of controls: they achieve this simply by sampling with $p(x_t|x_{t-1}, u_t)$. On the other hand, KBF filter must learn the transition model $p(x_t|x_{t-1}, u_t)$; this can be harder than learning the transition model $p(x_t|x_{t-1})$



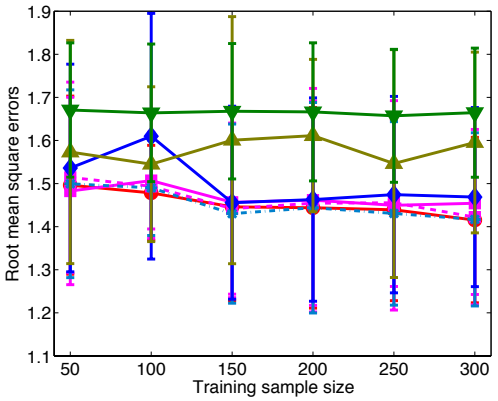

(a) RMSE (SSM 1a)




(b) RMSE (SSM 2a)




(c) RMSE (SSM 3a)

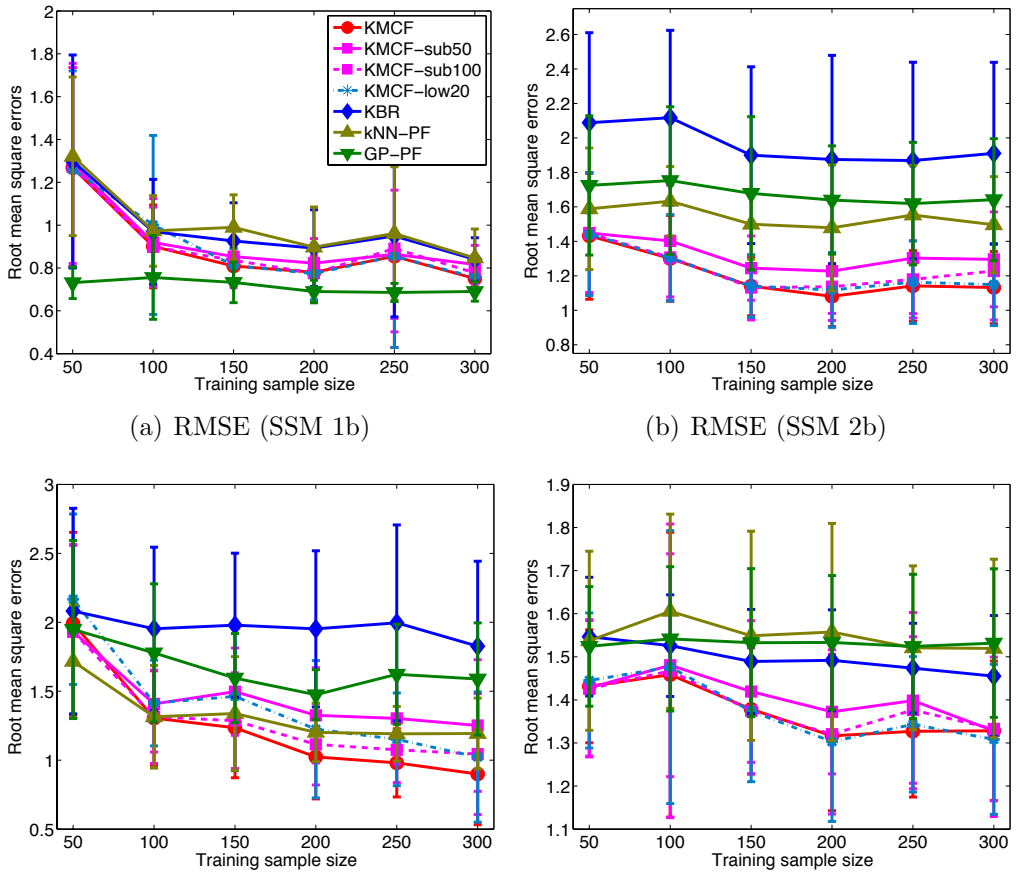



(d) RMSE (SSM 4a)

Figure 4.3: RMSE of the synthetic experiments in Section 4.5.1. The state-space models of these figures have no control in their transition models.

that has no control input.

We next compare computation time (Figure 4.5). KMCF was competitive or even slower than the KBR filter. This is due to the resampling step in KMCF. The speeding up methods (KMCF-low10, KMCF-low20, KMCF-sub50 and KMCF-sub100) successfully reduced the costs of KMCF. KMCF-low10 and KMCF-low20 scaled linearly to the sample size $n$; this matches the fact that Algorithm 5 reduces the costs of Kernel Bayes' Rule to $O(nr^2)$. On the other hand, the costs of KMCF-sub50 and KMCF-sub100 remained almost the same amounts over the difference sample sizes. This is because they reduce the sample size itself from $n$ to $r$, so the costs are reduced to $O(r^3)$ (see Algorithm 6). KMCF-sub50 and KMCF-sub100 are competitive to kNN-PF, which is fast as it only needs kNN searches to deal with




(a) RMSE (SSM 1b)

(b) RMSE (SSM 2b)

(c) RMSE (SSM 3b)

(d) RMSE (SSM 4b)

Figure 4.4: RMSE of synthetic experiments in Section 4.5.1. The state-space models of these figures include control $u_t$ in their transition models.

the training sample $\{(X_i, Y_i)\}_{i=1}^n$. In Figure 4.3 and 4.4, KMCF-low20 and KMCF-sub100 produced the results competitive to KMCF for SSMs {1a, 2a, 4a, 1b, 2b, 4b}. Thus for these models, such methods reduce the computational costs of KMCF without loosing much accuracy. KMCF-sub50 was slightly worse than KMCF-100. This indicates that the number of subsamples cannot be reduced to this extent if we wish to maintain the accuracy. For SSM 3a and 3b, the performance of KMCF-low20 and KMCF-sub100 were worse than KMCF, in contrast to the performance for the other models. The difference of SSM 3a and 3b from the other models is that the observation space is 10-dimensional: $\mathcal{Y} = \mathbb{R}^{10}$. This suggests that if the dimension is high, $r$ needs to be large to maintain the accuracy (recall that $r$ is the rank of low rank matrices in Algorithm 5, and the number of subsamples in Algorithm 6). This

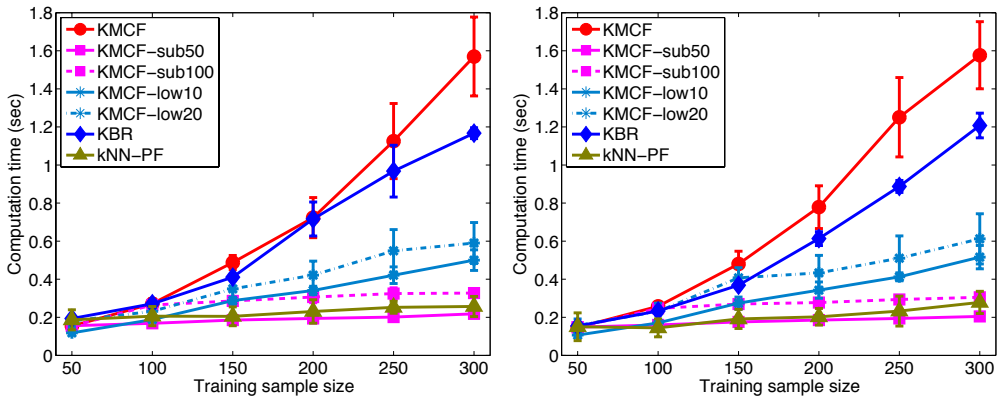



Figure 4.5: Computation time of synthetic experiments in Section 4.5.1. Left: SSM 1a. Right: SSM 1b.

is also implied by the experiments in the next subsection.

### 4.5.2 Vision-based mobile robot localization

We applied KMCF to the problem of vision-based mobile robot localization (Vlassis et al., 2002; Wolf et al., 2005; Quigley et al., 2010). We consider a robot moving in a building. The robot takes images with its vision camera as it moves. Thus the vision images form a sequence of observations $y_1, \ldots, y_T$ in time series; each $y_t$ is an image. On the other hand, the robot does not know its positions in the building; we define state $x_t$ as the robot's position at time $t$. The robot wishes to estimate its position $x_t$ from the sequence of its vision images $y_1, \ldots, y_t$. This can be done by filtering, i.e., by estimating the posteriors $p(x_t|y_1, \ldots, y_t)$ $(t = 1, \ldots, T)$. This is the robot localization problem. It is fundamental in robotics, as a basis for more involved applications such as navigation and reinforcement learning (Thrun et al., 2005).

The state-space model is defined as follows: the observation model $p(y_t|x_t)$ is the conditional distribution of images given position, which is very complicated and considered unknown. We need to assume position-image examples $\{(X_i, Y_i)\}_{i=1}^n$; these samples are given in the dataset described below. The transition model $p(x_t|x_{t-1}) := p(x_t|x_{t-1}, u_t)$ is the conditional distribution of the current position given the previous one. This involves a control input $u_t$ that specifies the movement of the robot. In the dataset we use, the control is given as odometry measurements. Thus we define $p(x_t|x_{t-1}, u_t)$ as the *odometry motion model*, which is fairly standard in robotics (Thrun et al., 2005). Specifically, we used the algorithm described in Table 5.6 of Thrun et al. (2005), with all of its parameters fixed to 0.1. The prior $p_{\text{init}}$ of the initial position $x_1$ is defined as a uniform distribution over the samples $X_1, \ldots, X_n$ in

$\{(X_i, Y_i)\}_{i=1}^n$.

As a kernel $k_{\mathcal{Y}}$ for observations (images), we used the Spatial Pyramid Matching Kernel of Lazebnik et al. (2006). This is a positive definite kernel developed in the computer vision community, and is also fairly standard. Specifically, we set the parameters of this kernel as suggested in Lazebnik et al. (2006): this gives a 4200 dimensional histogram for each image. We defined the kernel $k_{\mathcal{X}}$ for states (positions) as Gaussian. Here the state space is the 4-dimensional space: $\mathcal{X} = \mathbb{R}^4$: two dimensions for location, and the rest for the orientation of the robot.[3]

The dataset we used is the COLD database (Pronobis and Caputo, 2009), which is publicly available. Specifically, we used the dataset *Freiburg, Part A, Path 1, cloudy*. This dataset consists of three similar trajectories of a robot moving in a building, each of which provides position-image pairs $\{(x_t, y_t)\}_{t=1}^T$. The length of each trajectory is about 70 meters. We used two trajectories for training and validation, and the rest for test. We made state-observation examples $\{(X_i, Y_i)\}_{i=1}^n$ by randomly subsampling the pairs in the trajectory for training. Note that the difficulty of localization may depend on the time interval (i.e., the interval between $t$ and $t-1$ in sec.) Therefore we made three test sets (and training samples for state transitions in KBR filter) with different time intervals: 2.27 sec. ($T = 168$), 4.54 sec. ($T = 84$) and 6.81 sec. ($T = 56$).

In these experiments, we compared KMCF with three methods: kNN-PF, KBR filter, and the naive method (NAI) defined below. For KBR filter, we also defined the Gaussian kernel on the control $u_t$, i.e., on the difference of odometry measurements at time $t-1$ and $t$. The naive method (NAI) estimates the state $x_t$ as a point $X_j$ in the training set $\{(X_i, Y_i)\}$ such that the corresponding observation $Y_j$ is closest to the observation $y_t$. We performed this as a baseline. We also used the Spatial Pyramid Matching Kernel for these methods (for kNN-PF and NAI, as a similarity measure of the nearest neighbors search). We did not compare with GP-PF, since it assumes that observations are real vectors and thus cannot be applied to this problem straightforwardly. We determined the hyper-parameters in each method by cross validation. To reduced the cost of the resampling step in KMCF, we used the method discussed in Section 3.3 with $\ell = 100$. The low rank approximation method (Algorithm 5) and the subsampling method (Algorithm 6) were also applied to reduce the computational costs of KMCF. Specifically, we set $r = 50, 100$ for Algorithm 5 (described as KMCF-low50 and KMCF-low100 in the results below), and $r = 150, 300$ for Algorithm 6 (KMCF-sub150 and KMCF-sub300).

Note that in this problem, the posteriors $p(x_t|y_{1:t})$ can be highly multimodal. This is because similar images appear in distant locations. Therefore the posterior mean $\int x_t p(x_t|y_{1:t})dx_t$ is not appropriate for point estimation of the ground-truth position $x_t$. Thus for KMCF and KBR filter, we employed the heuristic for mode estimation

[3] We projected the robot's orientation in $[0, 2\pi]$ onto the unit circle in $\mathbb{R}^2$.

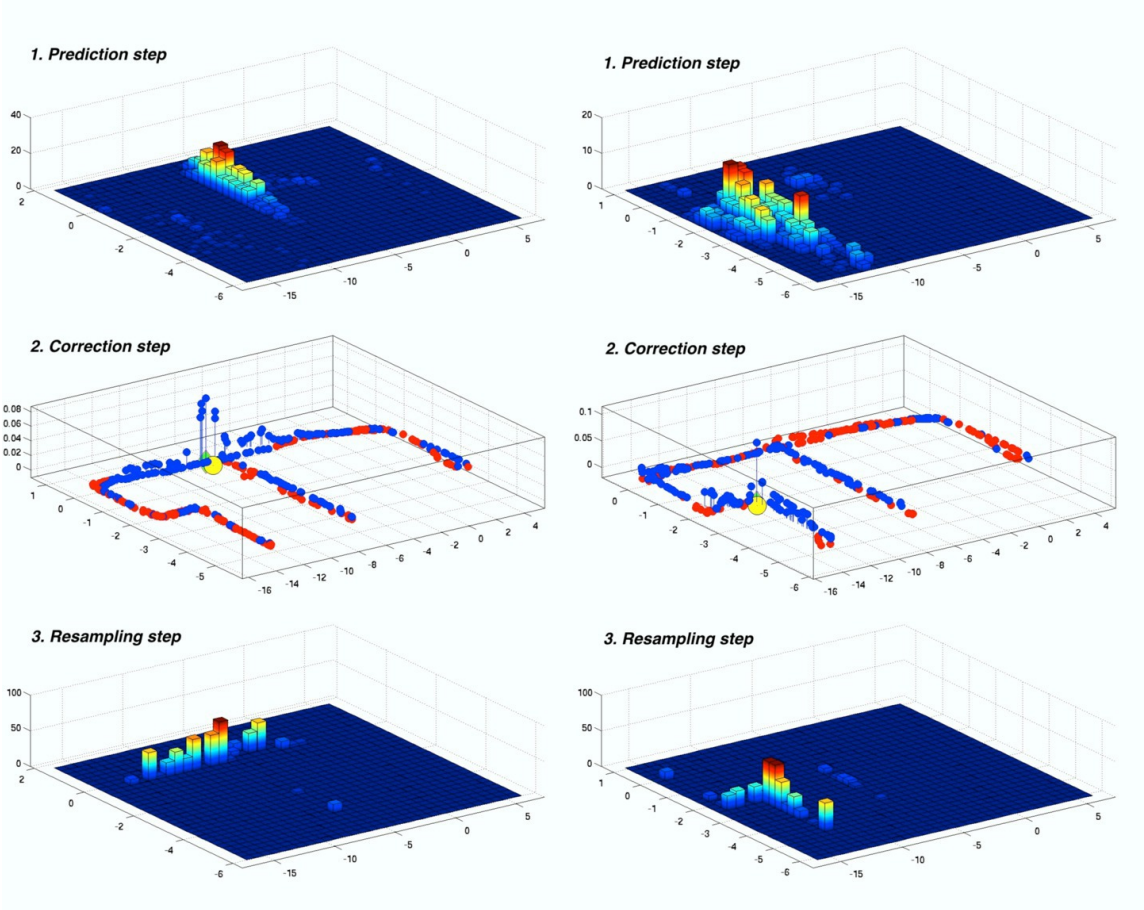
explained in Section 4.2.4. For kNN-PF, we used a particle with maximum weight for the point estimation. We evaluated the performance of each method by RMSE of location estimates. We ran each experiment 20 times for each training set of different size.

**Results.** First, we demonstrate the behaviors of KMCF with this localization problem. Figures 4.6 and 4.7 show iterations of KMCF with $n = 400$, applied to the test data with time interval 6.81 sec. The units of each axis are in meters. Figure 4.6 illustrates iterations that produced accurate estimates, while Figure 4.7 describes situations where location estimation is difficult.

Figures 4.8 and 4.9 show the results in RMSE (in meters) and computational time, respectively. For all the results KMCF and that with the computational reduction methods (KMCF-low50, KMCF-low100, KMCF-sub150 and KMCF-300) performed better than KBR filter. These results show the benefit of directly manipulating the transition models with sampling. KMCF was competitive with kNN-PF for the interval 2.27 sec.; note that kNN-PF was originally proposed for the robot localization problem. For the results with the longer time intervals (4.54 sec. and 6.81 sec.), KMCF outperformed kNN-PF.

We next investigate the effect on KMCF of the methods to reduce computational cost. The performance of KMCF-low100 and KMCF-sub300 are competitive with KMCF; those of KMCF-low50 and KMCF-sub150 degrade as the sample size increases. Note that $r = 50, 100$ for Algorithm 5 are larger than those in Section 4.5.1, though the values of the sample size $n$ are larger than those in Section 4.5.1. Also note that the performance of KMCF-sub150 is much worse than KMCF-sub300. These results indicate that we may need large values for $r$ to maintain the accuracy for this localization problem. Recall that the Spatial Pyramid Matching Kernel gives essentially a high-dimensional feature vector (histogram) for each observation. Thus the observation space $\mathcal{Y}$ may be considered high-dimensional. This supports the hypothesis in Section 4.5.1 that if the dimension is high, the computational cost reduction methods may require larger $r$ to maintain accuracy.

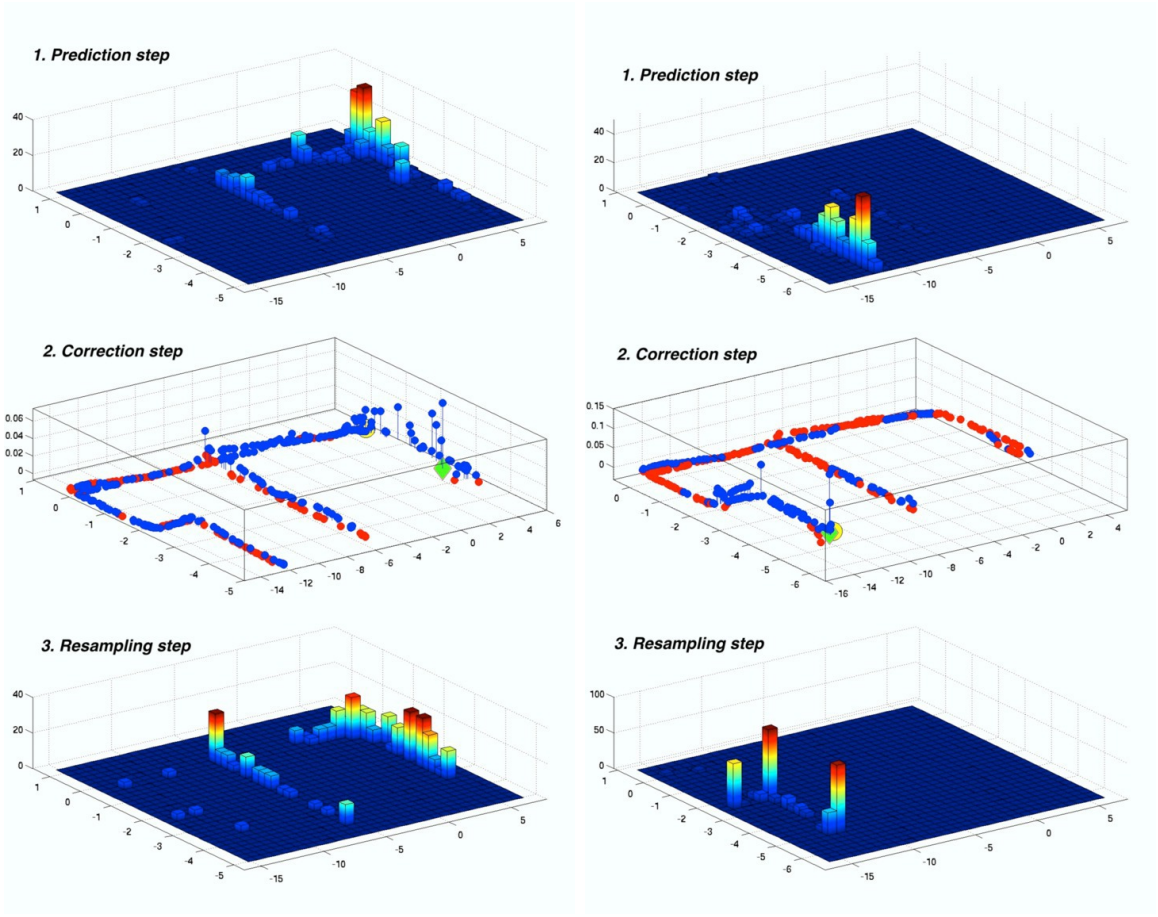
Finally, let us look at the results in computation time (Figure 4.9). The results are similar to those in Section 4.5.1. Even though the values for $r$ are relatively large, Algorithm 5 and Algorithm 6 successfully reduced the computational costs of KMCF.




(a) $t = 29$. $\|\hat{x}_t - x_t\| = 0.26378$.

(b) $t = 43$. $\|\hat{x}_t - x_t\| = 0.26315$.

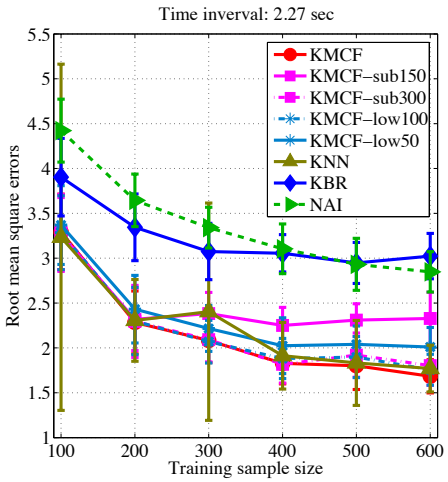
Figure 4.6: Demonstration results. Each column corresponds to one iteration of KMCF. Top (prediction step): histogram of samples for prior. Middle (correction step): weighted samples for posterior. The blue and red stems indicate positive and negative weights, respectively. The yellow ball represents the ground-truth location $x_t$, and the green diamond the estimated one $\hat{x}_t$. Bottom (resampling step): histogram of samples given by the resampling step.



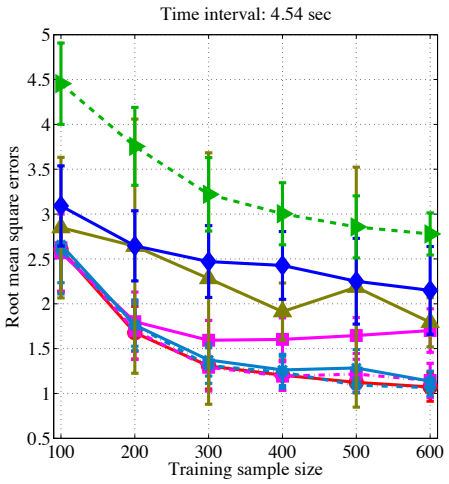

(a) $t = 11$. $\|\hat{x}_t - x_t\| = 2.3443$.

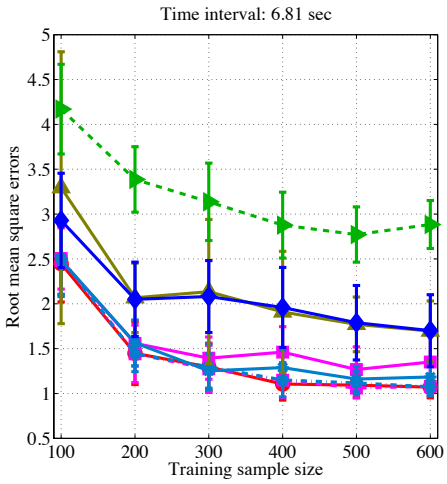
(b) $t = 40$. $\|\hat{x}_t - x_t\| = 0.3273$.

Figure 4.7: Demonstration results (see also the caption of Figure 4.6). Here we show time points where observed images are similar to those in distant places. Such a situation often occurs at corners, and makes location estimation difficult. (a) The prior estimate is reasonable, but the resulting posterior has modes in distant places. This makes the location estimate (green diamond) far from the true location (yellow ball). (b) While the location estimate is very accurate, modes also appear at distant locations.



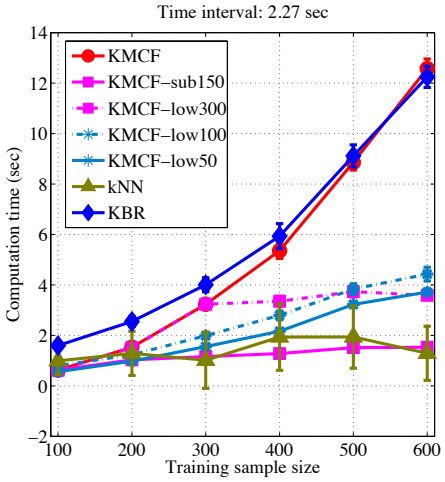

(a) RMSE (time interval: 2.27 sec; $T = 168$)



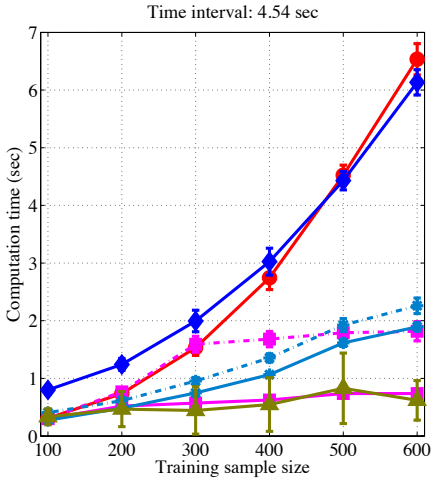




(b) RMSE (time interval 4.54 sec; $T = 84$) (c) RMSE (time interval 6.81 sec; $T = 56$)

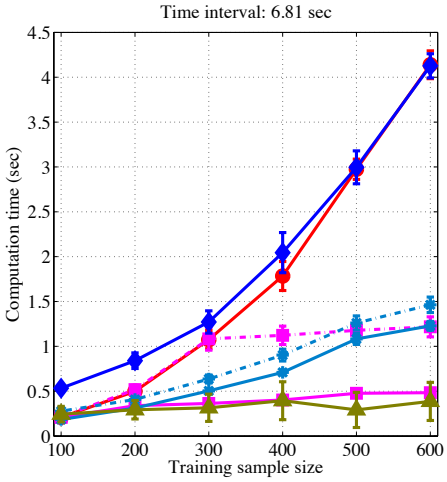
Figure 4.8: RMSE of the robot localization experiments in Section 4.5.2. (a), (b) and (c) show the cases for time interval 2.27 sec. , 4.54 sec. and 6.81 sec., respectively.




(a) Computation time (sec.) ($T = 168$)




(b) Computation time (sec.) ($T = 84$)




(c) Computation time (sec.) ($T = 56$)

Figure 4.9: Computation time of the localization experiments in Section 4.5.2. (a), (b) and (c) show the cases for time interval 2.27 sec. , 4.54 sec. and 6.81 sec., respectively. Note that the results show the run time of each method.

# Chapter 5

# Decoding distributions from empirical kernel means

Let us consider a kernel mean $m_P := \int k(\cdot, x)dP(x)$ of a distribution $P$ with $k$ being a characteristic kernel on a measurable space $\mathcal{X}$. As we have seen, in general the kernel mean is estimated in the form of a weighted sum of feature vectors:

$$\hat{m}_P := \sum_{i=1}^{n} w_i k(\cdot, X_i), \tag{5.1}$$

with some weights $w_1, \ldots, w_n \in \mathbb{R}$ and samples $X_1, \ldots, X_n \in \mathcal{X}$. Suppose that this estimate is accurate, i.e., the error $\|\hat{m}_P - m_P\|_{\mathcal{H}}$ is small, where $\mathcal{H}$ denotes the RKHS of the kernel $k$. Then the estimate (5.1) would provide accurate information about the distribution $P$, since the kernel mean $m_P$ maintains all information about $P$.

The aim of this chapter is to investigate a way of decoding the information of the distribution $P$ from a kernel mean estimate $\hat{m}_P$. This is an important problem for the theory and practice of kernel mean embeddings. For example, recall the application of kernel mean embeddings to a state-space model discussed in Chapter 4. In this application, the distribution $P$ may be a posterior distribution of a state variable at a certain time. Then the algorithm of Chapter 4 outputs an estimate for the kernel mean $m_P$ of the posterior in the form of (5.1). However, just having the estimate $\hat{m}_P$ is not enough, since the goal of filtering is to estimate the posterior $P$ itself. Therefore we need a method for decoding the information of $P$ from the kernel mean estimate $\hat{m}_P$. The same problems also appears in other applications (Song et al., 2013).

Typical examples of the information of $P$ to be decoded are its moments, such as the mean and covariance. Assume that we would like to estimate the mean $\int xdP(x)$. How can we estimate this quantity using the empirical kernel mean (5.1)? By regarding (5.1) as an empirical distribution, one might use the weighted average $\sum_{i=1}^{n} w_i X_i$. The question is whether this can be justified.

It is known that this way of estimating the expectation of a function is valid if the function belongs to the RKHS (see Chapter 2). That is, let $f \in \mathcal{H}$ be a function in the RKHS, and suppose we are interested in the expectation:

$$\mathbf{E}_{X \sim P}[f(X)] := \int f(x) dP(x). \tag{5.2}$$

Then this can be estimated by the weighed average:

$$\hat{\mathbf{E}}_{X \sim P}[f(X)] := \sum_{i=1}^{n} w_i f(X_i) \tag{5.3}$$

with weights $w_1, \ldots, w_n$ and samples $X_1, \ldots, X_n$ being those of the empirical kernel mean (5.1).

The question is whether this is also valid for functions $f$ *outside* the RKHS $\mathcal{H}$. For example, polynomial functions, whose expectations yield the moments, are not included in the Gaussian RKHS (Minh, 2010). Therefore if the kernel is Gaussian, the consistency of the weighted average (5.3) is not guaranteed for a polynomial function $f$. In other words, it is not justified the the moments of $P$ can be estimated by the form (5.3). The Gaussian RKHS also does not contain the indicator function on any subset $A \subset \mathbb{R}^d$, whose expectation (5.2) yields the probability measure $P(A)$ on that set. These facts are problematic, since the Gaussian kernel has been widely used in the literature on kernel mean embeddings (see, e.g., Song et al. (2013)).

Note that this problem is meaningful when we do not have access to an i.i.d. sample from $P$. If we have an i.i.d. sample from $P$, then the central limit theorem guarantees that the estimator (5.3) with uniform weights $w_1 = \cdots = w_n = 1/n$ converges to the expectation $\mathbf{E}_{X \sim P}[f(X)]$ at a rate $O_p(n^{-1/2})$. Therefore we are interested in situations where we do not have samples from $P$. These are situations where estimation of conditional distributions involve, such as inference in graphical models (Song et al., 2009; Fukumizu et al., 2013; Song et al., 2013) and reinforcement learning (Grünewälder et al., 2012b; Nishiyama et al., 2012; van Hoof et al., 2015).

In this chapter, we address the following problem: given the consistent kernel mean estimator (5.1), construct consistent estimators for quantities involved with $P$. Examples of such quantities include the moments (e.g., mean and variance) of $P$, the probability mass $P(A)$ on some measurable set $A \subset \mathcal{X}$, and density values of $P$. These quantities are often of practical interest, since they can be used in prediction. Therefore it is very important to discuss how to estimate such quantities using the kernel mean estimator (5.1). Recall that (5.1) does not directly provide us the information of the distribution $P$, since the kernel mean $m_P$ is not the distribution itself, but its representation in the RKHS.

We address this problem by extending the theoretical guarantee of the estimator

(5.3) to functions $f$ that lie *outside* the RKHS. Specifically, we focus on the case of the Gaussian RKHS on $\mathcal{X} = \mathbb{R}^d$, and show that (5.3) can be consistent for functions in the *Besov* space. The Besov space contains a broad class of functions including those in the Gaussian RKHS. It is a generalization of the Sobolev space, and contains functions with smoothness of a certain degree. Importantly, the polynomial and indicator functions are contained in the Besov space under certain assumptions. Therefore we can guarantee that the estimator (5.3) can be used to estimate the moments and probability masses, even when the kernel is Gaussian.

We also prove that the estimator (5.3) can be used to estimate the density of $P$, if we replace $f$ in (5.3) by *a smoothing kernel*. A smoothing kernel $J : \mathbb{R}^d \to \mathbb{R}$ is a function satisfying the properties of a probability density: $J(x) \geq 0$ and $\int J(x)dx = 1$. It has been used in classical kernel density estimation, and is a concept different from positive definite kernels. Let $p(x)$ be the density value of $P$ at $x \in \mathcal{X}$. Then this can be estimated as

$$\hat{p}(x) := \frac{1}{h^d} \sum_{i=1}^{n} w_i J\left(\frac{X_i - x}{h}\right), \tag{5.4}$$

where $h > 0$ is a bandwidth selected carefully. We prove that (5.4) converges to the true density $p(x)$, as the kernel mean estimates converge to the true kernel mean and $h$ goes to 0 at a appropriate rate. The proof is also based on the arguments using the Besov space, since the smoothing kernel is not contained in the Gaussian RKHS in general.

This chapter proceeds as follows. We review related works in Section 5.1. Here in particular, we discuss the literature on kernel mean embeddings to which our results can be applied. In Section 5.2, we introduce the Besov space and discuss its relations to other function spaces such as the Sobolev space and the Gaussian RKHS. We present our main result using the Besov space in Section 5.3, and the result on density estimation in Section 5.4. Simulation results are presented in Section 5.5. We collect all proofs in Section 5.6.

## 5.1 Related work

We first review existing methods that have been used for decoding the information of distributions from kernel mean estimates.

**Pre-image computation.** A popular method is to compute the *pre-image* of the kernel mean estimate $\hat{m}_P$: the pre-image is a point in the original space such that

$$x_{\mathrm{pre}} := \arg\min_{x \in \mathcal{X}} \|k(\cdot, x) - \hat{m}_P\|_{\mathcal{H}}. \tag{5.5}$$

Namely, the pre-image $x_{\mathrm{pre}}$ is the point whose feature vector $k(\cdot, x_{\mathrm{pre}})$ is closest to the mean estimate $\hat{m}_P$. The pre-image may be interpreted as a point that represents the distribution $P$, and has been widely used for the purpose of prediction (Song et al., 2009, 2010a; Fukumizu et al., 2013; Song et al., 2013).

As pointed out by Song et al. (2009, 2010a), the pre-image may be regarded as the mode of the density of $P$, when the kernel is Gaussian. This can be seen as follows. The square of the objective function in (5.5) can be expanded as

$$\begin{aligned}\arg\min_{x\in\mathcal{X}} \|k(\cdot,x)-\hat{m}_P\|_{\mathcal{H}}^2 &= \arg\min_{x\in\mathcal{X}} k(x,x) - 2\langle k(\cdot,x), \hat{m}_P\rangle_{\mathcal{H}} + \|\hat{m}_P\|_{\mathcal{H}}^2 \\ &= \arg\min_{x\in\mathcal{X}} k(x,x) - 2\hat{m}_P(x) \\ &= \arg\max_{x\in\mathcal{X}} -k(x,x) + 2\hat{m}_P(x).\end{aligned}$$

Therefore when $k(x,x) = C$ for some $C > 0$ for all $x \in \mathcal{X}$ (e.g., when $k$ is Gaussian), the pre-image is

$$x_{\mathrm{pre}} = \arg\max_{x\in\mathcal{X}} \hat{m}_P(x).$$

Let $k$ be the Gaussian kernel with bandwidth $\gamma > 0$: $k(x,x') := k_\gamma(x,x') := \exp(-\|x-x'\|^2/\gamma^2)$. Then

$$x_{\mathrm{pre}} = \arg\max_{x\in\mathcal{X}} \sum_{i=1}^n w_i k_\gamma(x, X_i).$$

Thus $x_{pre}$ may be seen as the maximum of a (weighted) kernel density estimator. Note that in kernel density estimation, the bandwidth should decrease as the sample size increases. On the other hand, the bandwidth in the positive definite kernel is fixed regardless of the sample size. Therefore the above density estimator is biased, in a sense that it does not decrease the bandwidth. Empirically, the pre-image have been shown to work well in some applications. This would be because the bandwidth of the Gaussian kernel as a positive definite kernel was set so that the above density estimator performs well.

**Density estimation.** As discussed above, the kernel mean estimate $\hat{m}_P(x)$ may be seen as a biased estimate of the density of the distribution $P$, when the kernel is Gaussian. Song and Dai (2013); Song et al. (2014) use this interpretation for the purpose of density estimation. Later we will show that if we use a smoothing kernel with degreasing bandwidths in the place of the Gaussian kernel, it will be a consistent density estimator. This explains why these work well in practice.

**Moments (mean).** For the purpose of prediction, we often wish to estimate the mean of the distribution $P$. In the works by Boots et al. (2013, 2014); Zhu et al. (2014); Kanagawa et al. (2014), the mean is estimated using a kernel mean estimate. The mean can be given as the expectations of the coordinate projection functions $f_j(x) = x_j$ $(j = 1, \ldots, d)$: $\int x dP(x) = (\mathbf{E}_{X\sim P}[f_j(X)])_{j=1}^d$. Therefore it can be estimated with the kernel mean estimate $\hat{m}_P$ as

$$\sum_{i=1}^n w_i X_i = (\sum_{i=1}^n w_i f_j(X_i))_{j=1}^d.$$

However, the coordinate projection functions may not be included in widely used RKHSs such as the Gaussian RKHS. To alleviate this problem, Boots et al. (2013); Zhu et al. (2014) proposed to approximate these functions by regression, so that the resulting approximate functions are included in the RKHS.

**Fitting Gaussian mixtures.** There is another method for estimating the density of $P$ from a kernel mean estimate (Song et al., 2008; McCalman et al., 2013). This method models the density as a Gaussian mixture, and then embeds it into the RKHS. The parameters of the mixture is then learned, by minimizing the RKHS distance between the embedded mixture and the kernel mean estimate.

**Reinforcement learning.** In applications to reinforcement learning (Grünewälder et al., 2012b; Nishiyama et al., 2012; van Hoof et al., 2015), one needs to compute the expectation of a value function. However, the value function may not be included in an RKHS in general. Therefore our result may be beneficial to these works, as it guarantees that the expectation can be computed even when the value function is not included in the RKHS.

## 5.2 Function spaces

Here we review function spaces on the Euclidian space $\mathbb{R}^d$ and relations between them. These serve as a basis for our analysis.

**Notation.** In this chapter we follow the notation of Eberts and Steinwart (2013). For $\alpha \in \mathbb{R}$, we denote by $\lfloor \alpha \rfloor \in \mathbb{Z}$ the greatest integer smaller or equal to $\alpha$. For a measure $\nu$ on $\mathbb{R}^d$ and a constant $p \in (0, \infty]$, $L_p(\nu)$ denotes the Banach space of $p$-integrable functions with respect to $\nu$. If $\nu$ is the Lebesgue measure on $\mathcal{X} \subset \mathbb{R}^d$, we write $L_p(\mathcal{X}) := L_p(\nu)$. We denote by $\mu$ the Lebesgue measure on $\mathcal{X} \subset \mathbb{R}^d$.

### 5.2.1 Sobolev spaces

Here we introduce Sobolev spaces (Adams and Fournier, 2003). These spaces have close relation to Besov spaces defined in the next subsection, which are main tools of our analysis. Our main motivation to introduce the Sobolev spaces is to provide intuition about the Besov spaces.

Let $\mathbb{N}_0^d$ denote the $d$ dimensional space of non-negative integers, $\alpha := (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$ be a multi-index, and $|\alpha| := \sum_{i=1}^n \alpha_i$. Denote by $\partial^{(\alpha)} f$ the $\alpha$-th weak derivative of a function $f$.

**Definition 1.** *Let $m \in \mathbb{N}$ and $1 \leq p \leq \infty$ be constants, and $\nu$ be a measure on $\mathbb{R}^d$. Then the Sobolev of order $m$ with respect to $\nu$ is defined by*

$$W_p^m(\nu) := \left\{ f \in L_p(\nu) : \partial^{(\alpha)} f \in L_p(\nu) \text{ exists for all } \alpha \in \mathbb{N}_0^d \text{ with } |\alpha| \leq m \right\}. \quad (5.6)$$

*Moreover, the norm is defined by*

$$\|f\|_{W_p^m(\nu)} := \left( \sum_{|\alpha| \leq m} \|\partial^{(\alpha)} f\|_{L_p(\nu)}^p \right)^{1/p}. \quad (5.7)$$

*If $\nu$ is the Lebesgue measure on $\mathcal{X} \subset \mathbb{R}^d$, we define $W_p^m(\mathcal{X}) := W_p^m(\nu)$.*

The Sobolev space (5.6) consists of functions in $L_p(\nu)$ whose weak derivatives up to order $m$ exist and are contained in $L_p(\nu)$. Thus the order $m$ quantifies the smoothness of functions: functions in $W_p^m(\nu)$ get smoother as $m$ increases. In fact, from the definition, the following is immediate:

$$W_p^{m'}(\nu) \subset W_p^m(\nu), \quad m \leq m'.$$

The Sobolev space is a Banach space with respect to the norm (5.7). For the case $p = 2$, it becomes a Hilbert space with respect to the inner product that induces the norm (5.7).

### 5.2.2 Besov spaces

We now introduce Besov spaces; for details, we refer to (Adams and Fournier, 2003, Chapter 7) and (DeVore and Lorentz, 1993, Chapter 2). There are several ways to define the Besov spaces. Following Eberts and Steinwart (2013), we define them via the quantity called *modulus of smoothness* (DeVore and Lorentz, 1993, Chapter 2) (Eberts and Steinwart, 2013, Definition 2.1.).

To this end, we first define the notion of higher order differences (DeVore and Lorentz, 1993, p.44).

**Definition 2** (Higher order differences)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ be a subset with nonempty interior, $\nu$ be a measure on $\mathcal{X}$, and $f \in L_p(\nu)$ with $p \in (0, \infty]$. Let $r \in \mathbb{N}$ and $h \in [0, \infty)^d$. Then the $r$-th difference of $f$ with respect to $h$ is a function $\Delta_h^r(f, \cdot) : \mathcal{X} \to \mathbb{R}$ defined by*

$$\Delta_h^r(f, x) = \begin{cases} \sum_{j=0}^r \binom{r}{j} (-1)^{r-j} f(x + jh) & \text{if } x \in \mathcal{X}_{r,h} \\ 0 & \text{if } x \notin \mathcal{X}_{r,h} \end{cases} \tag{5.8}$$

*where $\mathcal{X}_{r,h} := \{x \in \mathcal{X} : x + sh \in \mathcal{X},\ \forall s \in [0, r]\}$.*

Alternatively, higher order differences may be defined by induction (DeVore and Lorentz, 1993, p.44, Eq. (7.1)):

$$\Delta_h^r(f, x) = \begin{cases} \Delta_h^1(\Delta_h^{r-1}(f, \cdot), x) & \text{if } x \in \mathcal{X}_{r,h} \\ 0 & \text{if } x \notin \mathcal{X}_{r,h} \end{cases}, \quad r \geq 2 \tag{5.9}$$

with $\Delta_h^1(f, x) = f(x + h) - f(x)$.

We now introduce the modulus of smoothness.

**Definition 3** (Modulus of smoothness)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ be a subset with nonempty interior, $\nu$ be a measure on $\mathcal{X}$, and $f \in L_p(\nu)$ with $p \in (0, \infty]$. For $r \in \mathbb{N}$, the $r$-the modulus of smoothness of $f$ is a function $\omega_{r,L_p(\nu)}(f, \cdot) : [0, \infty) \to [0, \infty)$ defined by*

$$\omega_{r,L_p(\nu)}(f, t) = \sup_{\|h\|_2 \leq t} \|\Delta_h^r(f, \cdot)\|_{L_p(\nu)} \quad (t \geq 0), \tag{5.10}$$

*where $\Delta_h^r(f, \cdot)$ is defined by (5.8).*

Based on the modulus of smoothness, we now define the Besov space.

**Definition 4** (Besov space)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ be a subset with nonempty interior and $\nu$ be a measure on $\mathcal{X}$. Let $p, q \in [1, \infty]$, $\alpha \in (0, \infty)$ and $r := \lfloor \alpha \rfloor + 1$. The Besov space $B_{p,q}^\alpha(\nu)$ is a Banach space of functions given by*

$$B_{p,q}^\alpha(\nu) := \{f \in L_p(\nu) : |f|_{B_{p,q}^\alpha(\nu)} < \infty\}, \tag{5.11}$$

*where $|f|_{B_{p,q}^\alpha(\nu)}$ is the semi norm defined as*

$$|f|_{B_{p,q}^\alpha(\nu)} = \begin{cases} \left(\int_0^\infty (t^{-\alpha} \omega_{r,L_p(\nu)}(f, t))^q \frac{dt}{t}\right)^{1/q} & 1 \leq q < \infty. \\ \sup_{t>0} \left(t^{-\alpha} \omega_{r,L_p(\nu)}(f, t)\right) & q = \infty, \end{cases} \tag{5.12}$$

*where $\omega_{r,L_p(\nu)}(f, \cdot)$ is given by (5.10). The norm is defined as*

$$\|f\|_{B_{p,q}^\alpha(\nu)} := \|f\|_{L_p(\nu)} + |f|_{B_{p,q}^\alpha(\nu)}, \quad \forall f \in B_{p,q}^\alpha(\nu).$$

*If $\nu$ is the Lebesgue measure on $\mathcal{X}$, we write $B^{\alpha}_{p,q}(\mathcal{X}) := B^{\alpha}_{p,q}(\nu)$.*

**Relation to Sobolev spaces.** The Besov spaces have close relation to the Sobolev spaces introduced in Section 5.2.1. For example, consider the Besov space $B^{m}_{p,q}(\mathbb{R}^d)$ with $m \in \mathbb{N}$, $p \in (1, \infty)$, and $\max(p, 2) \leq q \leq \infty$. Then this Besov space contains the corresponding Sobolev space (Edmunds and Triebel, 1996, pp.25-27 and p.44):

$$W^{m}_{p}(\mathbb{R}^d) \subset B^{m}_{p,q}(\mathbb{R}^d). \tag{5.13}$$

In particular, for the case $p = q = 2$, these spaces are equivalent:

$$W^{m}_{2}(\mathbb{R}^d) = B^{m}_{2,2}(\mathbb{R}^d). \tag{5.14}$$

The inclusion (5.13) implies that the Besov space $B^{m}_{p,q}(\mathbb{R}^d)$ contains the following functions: (i) $m$-times continuously differentiable functions with compact supports; (ii) Gaussian functions $f(x) := A \exp(-B\|x - \mu\|^2)$ with any $A, B > 0$ and $\mu \in \mathbb{R}^d$.

**Useful inequality.** Let $f \in B^{\alpha}_{p,\infty}(\nu)$ be any function in the Besov space. By the definition of the semi norm (5.12) with $q = \infty$, the following holds for all $t > 0$:

$$\omega_{r,L_p(\nu)}(f, t) \leq |f|_{B^{\alpha}_{p,\infty}(\nu)} t^{\alpha}. \tag{5.15}$$

We will use this inequality in our proofs in Section 5.6.

### 5.2.3 Gaussian reproducing kenrel Hilbert spaces

Finally, we review the properties of Gaussian RKHSs and their relation to the above spaces. In particular, we focus on the relation to the Besov space $B^{\alpha}_{2,\infty}(\mathcal{X})$ with $\mathcal{X} \subset \mathbb{R}^d$, as this is the space we will use below.

Let $\mathcal{X} \subset \mathbb{R}^d$ be an arbitrary nonempty set, $\gamma > 0$ be a constant, and $k_\gamma : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be the Gaussian kernel with bandwidth $\gamma$, defined as

$$k_\gamma(x, x') := \exp\left(-\frac{\|x - x'\|^2}{\gamma^2}\right). \tag{5.16}$$

We denote by $\mathcal{H}_\gamma$ the RKHS associated with $k_\gamma$. Let $\langle \cdot, \cdot \rangle_{\mathcal{H}_\gamma}$ and $\| \cdot \|_{\mathcal{H}_\gamma}$ denote the inner-product and the norm of $\mathcal{H}_\gamma$, respectively.

Properties of the Gaussian RKHS have been extensively studied (Steinwart and Christmann, 2008, Section 4.4) (Minh, 2010). For example, functions in the Gaussian RKHS on $\mathcal{X} = \mathbb{R}^d$ can be characterized by their Fourier transforms (see e.g. Minh

(2010)):

$$\mathcal{H}_\gamma = \left\{ f \in C_0(\mathbb{R}^d) \cap L_2(\mathbb{R}^d) : \|f\|_{\mathcal{H}}^2 = \frac{\int_{\mathbb{R}^d} \exp(\frac{\gamma^2 \|\xi\|^2}{4}) |\hat{f}(\xi)|^2 d\xi}{(2\pi)^d (\gamma\sqrt{\pi})^d} < \infty \right\}, \quad (5.17)$$

where $C_0(\mathbb{R}^d)$ denotes the space of continuous functions on $\mathbb{R}^d$ and $\hat{f}$ the Fourier transform of $f$. Namely, the Gaussian RKHS consists of functions whose frequency spectrum decay exponentially fast. The characterization (5.17) follows from a general result on shift-invariant kernels; see (Wendland, 2005, Theorem 10.12). There are also characterizations based on explicit descriptions of orthonormal basis (Steinwart and Christmann, 2008, Section 4.4) (Minh, 2010).

Let $f \in L_2(\mathbb{R}^d)$ be a function, $\hat{f}$ be its Fourier transform, and $m \in \mathbb{N}$ any positive integer. It is well known (e.g. p. 252 of Adams and Fournier (2003)) that $f$ is included in the Sobolev space $W_2^m(\mathbb{R}^d)$ if and only if the function $u_m(\xi) := (1 + \xi)^{m/2} \hat{f}(\xi)$ is included in $L_2(\mathbb{R}^d)$. This property and (5.17) imply that the Gaussian RKHS is contained in the Sobolev space of any order $m \in \mathbb{N}$:

$$\mathcal{H}_\gamma \subset W_2^m(\mathbb{R}^d), \quad m \geq 1. \quad (5.18)$$

Therefore from (5.13), for any $2 \leq q \leq \infty$, we have

$$\mathcal{H}_\gamma \subset B_{2,q}^m(\mathbb{R}^d), \quad m \geq 1. \quad (5.19)$$

The characterization (5.17) also implies that as $\gamma$ gets larger, functions in $\mathcal{H}_\gamma$ becomes smoother. In fact, from (5.17) it is immediate that

$$\mathcal{H}_\gamma \subset \mathcal{H}_{\gamma'}, \quad 0 < \gamma' < \gamma.$$

The Gaussian RKHS $\mathcal{H}_\gamma$ consists of infinitely continuously differentiable functions, since the Gaussian kernel is infinitely continuously differentiable. This follows from Corollary 4.36 of Steinwart and Christmann (2008), which states that if a kernel is $m$-times continuously differentiable with $m \geq 0$, then RKHS functions are $m$-times continuously differentiable. In other words, RKHS functions inherit the smoothness of the kernel that defines the RKHS.

## 5.3 Main theorem

Let $P$ be a probability distribution on $\mathbb{R}^d$ and $m_P := \int k_\gamma(\cdot, x) dP(x) \in \mathcal{H}_\gamma$ be the kernel mean of $P$. Suppose that we are given a consistent estimator $\hat{m}_P$ of the kernel

mean $m_P$

$$\hat{m}_P := \sum_{i=1}^{n} w_i k_\gamma(\cdot, X_i) \tag{5.20}$$

such that

$$\lim_{n\to\infty} \|\hat{m}_P - m_P\|_{\mathcal{H}_\gamma} = 0.$$

Here the samples $X_1, \ldots, X_n \in \mathbb{R}^d$ are random variables, and the weights $w_1, \ldots, w_n \in \mathbb{R}$ are assumed to be given by some algorithm based on these samples and other random variables. Note that the samples $X_1, \ldots, X_n$ and the weights $w_1, \ldots, w_n$ in general depend on the sample size $n$, but we omit this dependency for notational simplicity.

For example, in the case of conditional mean embeddings (Song et al., 2009, 2013), the samples $X_1, \ldots, X_n$ are those from joint samples $(X_1, Y_1), \ldots, (X_n, Y_n)$, and the weights $w_1, \ldots, w_n$ are computed by linear algebraic operations on kernel matrices defined on these samples.

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function. In this section, we are interested in the estimation of the expectation $\mathbf{E}_{X\sim P}[f(X)] = \int f(x)dP(x)$. Specifically, we analyze the convergence behavior of the weighted sample estimator

$$\hat{E}[f(X)] := \sum_{i=1}^{n} w_i f(X_i)$$

where the weights $w_1 \ldots, w_n$ and the samples $X_1, \ldots, X_n$ are those of the kernel mean estimate (5.20). As mentioned, this is consistent if the function $f$ belongs to RKHS $\mathcal{H}_\gamma$. Our aim is to show that it can be also consistent if $f$ belongs to the Besov space $B^\alpha_{2,\infty}(\mathbb{R}^d)$ for some $\alpha > 0$. The following theorem provides a guarantee for this.

**Theorem 4.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function satisfying $f \in B^\alpha_{2,\infty}(\mathbb{R}^d) \cap L_\infty(\mathbb{R}^d)$ for some $\alpha > 0$. Let $P$ and $Q$ be probability distributions on $\mathbb{R}^d$ and assume that $P$ and $Q$ have density functions which belong to $L_\infty(\mathbb{R}^d)$. Let $\{(w_i, X_i)\}_{i=1}^n$ be such that $\sup_{\gamma>0} \mathbf{E}[\|\hat{m}_P - m_P\|_{\mathcal{H}_\gamma}] = O(n^{-b})$ and $\mathbf{E}[\sum_{i=1}^n w_i^2] = O(n^{-2c})$ as $n \to \infty$ for some $b \in (0, \infty)$ and $c \in (0, 1/2]$. Moreover, assume that $X_1, \ldots, X_n$ in (5.20) are i.i.d. with $Q$. Then we have*

$$\mathbf{E}\left[\left|\sum_{i=1}^{n} w_i f(X_i) - \mathbf{E}_{X\sim P}[f(X)]\right|\right] = O\left(n^{-\frac{2\alpha b - d(1/2-c)}{2\alpha+d}}\right) \quad (n \to \infty). \tag{5.21}$$

The rate (5.21) depends on the constants $\alpha$, $b$, $c$ and $d$. We provide discussions on the effects of these constants below:

**Smoothness of the function $\alpha$:** The constant $\alpha$ quantifies the smoothness of the

function $f \in B^{\alpha}_{2,\infty}(\mathbb{R}^d)$. As $\alpha$ increases, the rate (5.21) becomes faster. For example, suppose that the function $f$ belongs to $B^{\alpha}_{2,\infty}(\mathbb{R}^d)$ for arbitrarily large $\alpha > 0$, that is, $f$ is very smooth. Then the rate (5.21) becomes $O(n^{-b+\xi})$ for arbitrarily small $\xi > 0$. Namely in this case, (5.21) recovers the rate $O(n^{-b})$ of the case when $f$ belongs to the RKHS. Specifically, if $f \in \mathcal{H}_\gamma$, then $f \in B^{\alpha}_{2,\infty}(\mathbb{R}^d)$ for any $\alpha > 0$ and thus the rate (5.21) becomes $O(n^{-b+\xi})$.

**Rate of the kernel mean estimator $b$:** The constant $b$ comes from the convergence rate of the kernel mean estimator $O(n^{-b})$. It is reasonable to expect that the rate (5.21) becomes faster as $b$ increases.

**Effective sample size $c$:** The rate (5.21) gets faster as the constant $c$ increases. This constant comes from the assumption $\mathbf{E}[\sum_{i=1}^n w_i^2] = O(n^{-2c})$ on the quantity $\sum_{i=1}^n w_i^2$. This quantity can be understood as representing (the inverse of) the effective sample size (ESS) of the kernel mean estimator $\sum_{i=1}^n w_i k_\gamma(\cdot, X_i)$: ESS is defined as $ESS = 1/\sum_{i=1}^n w_i^2$, and roughly represents the number of samples that contribute to the estimation. Therefore the assumption $\mathbf{E}[\sum_{i=1}^n w_i^2] = O(n^{-2c})$ can be interpreted as requiring that the ESS increases as the sample size increases. For example, if the weights are uniform $w_1, \ldots, w_n = 1/n$, then we have $c = 1/2$ and thus the ESS is $n$. This assumption excludes situations where the weights are ill-behaved.

ESS is a notion common in the literature of particle methods (see e.g. Section 2.5.3 of Liu (2001) and Section 3.5 of Doucet and Johansen (2011)) and similar assumptions have been used for convergence analysis; see Definitions 1 and 2 of Douc and Moulines (2008).

**Dimensionality $d$:** The rate (5.21) gets slower as the dimensionality $d$ grows. This shows that even if the rate $O(n^{-b})$ of the given kernel mean estimator $\hat{m}_P$ does not depend on the dimensionality, the rate of the resulting estimator $\sum_{i=1}^n w_i f(X_i)$ does. Note that if $f$ belongs to the RKHS, then the rate of $\sum_{i=1}^n w_i f(X_i)$ becomes $O(n^{-b})$ and thus independent of the dimensionality. Therefore the dependence on the dimensionality comes from $f$ being outside of the RKHS.

We assumed that $X_1, \ldots, X_n$ are i.i.d. with some distribution $Q$. This is satisfied by various existing estimators (Song et al., 2013). For example, if $\hat{m}_P$ is given by the conditional embedding estimator (Song et al., 2009), then $X_1, \ldots, X_n$ are those from joint i.i.d. samples $\{(X_i, Y_i)\}$. Therefore in this case, $Q$ is the marginal distribution of the joint distribution of $\{(X_i, Y_i)\}$.

### 5.3.1 Expectations of infinitely differentiable functions

As a corollary of Theorem 4, we derive convergence rates for the expectations of infinitely differentiable functions. Specifically, if we consider certain polynomial functions, then their expectations yield the moments of the distribution $P$. Therefore the result below provides guarantees for the estimation of moments. Recall that this is not obvious beforehand, because polynomial functions are not included in the Gaussian RKHS (Minh, 2010).

Note that just assuming a function $f$ being infinitely differentiable is not sufficient to apply Theorem 4. This is because in general $f$ may not satisfy the assumption $f \in B^{\alpha}_{2,\infty}(\mathbb{R}^d) \cap L_{\infty}(\mathbb{R}^d)$. That is, $f$ may be neither square-integrable nor bounded on $\mathbb{R}^d$ (recall that $B^{\alpha}_{2,\infty}(\mathbb{R}^d)$ is a subspace of $L_2(\mathbb{R}^d)$). Therefore we need additional assumptions to obtain the result of Theorem 4 for infinitely differentiable functions.

Here we assume that the supports of the distributions $P$ and $Q$ are bounded. Under this assumption, we can relax the assumption $f \in B^{\alpha}_{2,\infty}(\mathbb{R}^d) \cap L_{\infty}(\mathbb{R}^d)$ to

$$f \in B^{\alpha}_{2,\infty}(B_R) \cap L_{\infty}(B_R), \tag{5.22}$$

where $B_R = \{x \in \mathbb{R}^d : \|x\| < R\}$ is an open ball with radius $R > 0$ that contains the supports of $P$ and $Q$. This is done by applying Stein's extension theorem (Stein, 1970, pp.180-192) (Adams and Fournier, 2003, p.154 and p.230). We then have the following proposition.

**Proposition 1.** *Assume that $P$, $Q$, and $\{(w_i, X_i)\}_{i=1}^n$ satisfy the conditions of Theorem 4. Moreover, assume that the supports of $P$ and $Q$ are bounded. Let $f : \mathbb{R}^d \to \mathbb{R}$ be an infinitely continuously differentiable function. Then for any $\xi > 0$, we have*

$$\mathbf{E}\left[\left|\sum_{i=1}^{n} w_i f(X_i) - E_P[f(X)]\right|\right] = O\left(n^{-b+\xi}\right) \ (n \to \infty).$$

Below we show some specific examples of Proposition 1. The first example is $f$ being a constant. More specifically, we consider the function $f(x) = 1$ for all $x \in \mathbb{R}^d$. This yields the following.

**Corollary 4.** *Assume that $P$, $Q$, and $\{(w_i, X_i)\}_{i=1}^n$ satisfy the conditions of Theorem 4. Moreover, assume that the supports of $P$ and $Q$ are bounded. Then for any $\xi > 0$, we have*

$$\mathbf{E}\left[\left|\sum_{i=1}^{n} w_i - 1\right|\right] = O\left(n^{-b+\xi}\right) \ (n \to \infty).$$

Corollary 4 shows that the sum of the weights $\sum_{i=1}^n w_i$ converges to 1 at a rate

arbitrarily close to the rate of the kernel mean estimator $\hat{m}_P = \sum_{i=1}^n w_i k_\gamma(\cdot, X_i)$. This provides a theoretical background for the normalization procedure of the filtering method in Chapter 4.

The next example is the mean of the distribution $\mu := \mathbf{E}_{X\sim P}[f(X)] \in \mathbb{R}^d$. We derive a convergence rate for the estimator $\hat{\mu} := \sum_{i=1}^n w_i X_i$. To this end, we consider the functions that output coordinates of an input variable: $f_i(x) := x_i,\ (i = 1, \ldots, d)$. Then we have $\mu_i = \mathbf{E}_{X\sim P}[f_i(X)]$ and $\hat{\mu}_i = \sum_{i=1}^n w_i f_i(X)$, where $\mu = (\mu_1, \ldots, \mu_d)^T$ and $\hat{\mu} = (\hat{\mu}_1, \ldots, \hat{\mu}_d)^T$. Observing that the functions $f_i$ are infinitely continuously differentiable, we have the following.

**Corollary 5.** *Assume that $P$, $Q$, and $\hat{m}_P$ satisfy the conditions of Theorem 4. Moreover, assume that the supports of $P$ and $Q$ are bounded. Then for any $\xi > 0$, we have*

$$\mathbf{E}\left[\left\|\sum_{i=1}^n w_i X_i - \mathbf{E}_{X\sim P}[X]\right\|\right] = O\left(n^{-b+\xi}\right)\ (n \to \infty).$$

In a similar manner, we can derive convergence rates for the estimation of (uncentered) moments.

### 5.3.2 Expectations of indicator functions on cubes

We next consider to estimate the probability measure $P(\Omega)$ of a measurable set $\Omega \subset \mathbb{R}^d$. To this end, note that $P(\Omega)$ can be written as the expectation of the indicator function defined as

$$I_\Omega(x) = \begin{cases} 1 & (x \in \Omega) \\ 0 & (x \notin \Omega) \end{cases}. \tag{5.23}$$

That is, we have $P(\Omega) = \mathbf{E}_{X\sim P}[I_\Omega(X)]$. Therefore we may define an estimator of $P(\Omega)$ as

$$\sum_{i=1}^n w_i I_\Omega(X_i) = \sum_{i:X_i\in\Omega} w_i. \tag{5.24}$$

Obviously we need some regularity conditions on the set $\Omega$. Here we consider the specific case of a cube, that is

$$\Omega := [a_1, b_1] \times \cdots [a_d, b_d] \subset \mathbb{R}^d,$$

with $-\infty < a_i < b_i < +\infty$ $(i = 1, \ldots, d)$. In this case, the measure $P(\Omega)$ could be useful for constructing credible intervals in Bayesian inference (Fukumizu et al., 2013). As we have $I_\Omega \in B^\alpha_{2,\infty}(\mathbb{R}^d)$ for any $0 < \alpha < 1/2$ in this case, we have the

following corollary.

**Corollary 6.** *Assume that $P$, $Q$, and $\{(w_i, X_i)\}_{i=1}^n$ satisfy the conditions in Theorem 4. Let $\Omega := [a_1, b_1] \times \cdots [a_d, b_d]$ with $-\infty < a_i < b_i < +\infty$ $(i = 1, \ldots, d)$. Assume that $b - d(1/2 - c) > 0$. Then for arbitrary small $\xi > 0$, we have*

$$\mathbf{E}\left[\left|\sum_{X_i \in \Omega} w_i - P(\Omega)\right|\right] = O\left(n^{-\frac{b-d(1/2-c)}{1+d}+\xi}\right).$$

The condition $b - d(1/2 - c) > 0$ of Corollary 6 is to guarantee that the exponent of the rate (5.25) is positive. For example, if the weights satisfy $\mathbf{E}[\sup_{i\in\{1,\ldots,n\}} |w_i|] = O(n^{-1})$, the condition is always satisfied.

## 5.4 Decoding density functions

Let $p$ be the density function of the distribution $P$. In this section, we show that the density $p$ can be estimated based on a consistent kernel mean estimator $\hat{m}_P = \sum_{i=1}^n w_i k_\gamma(\cdot, X_i)$. Specifically, we consider the estimator described as follows. Let $J : \mathbb{R}^d \to \mathbb{R}$ be a smoothing kernel satisfying the following conditions:

$$J(x) \geq 0, \quad \forall x \in \mathbb{R}^d, \tag{5.25}$$

$$\int J(x)dx = 1, \tag{5.26}$$

$$J \in B^\alpha_{2,\infty}(\mathbb{R}^d), \quad \forall \alpha > 0. \tag{5.27}$$

For example, this is satisfied if we let $J$ be the Gaussian density $J(x) := \frac{1}{\pi^d}\exp(-\|x\|^2)$. For a constant $h > 0$, we define a function $J_h : \mathbb{R}^d \to \mathbb{R}$ by

$$J_h(x) := J(x/h).$$

Then we define an estimator of the density $p(x_0)$ at any $x_0 \in \mathbb{R}^d$ as

$$\sum_{i=1}^n w_i J_h(X_i - x_0), \tag{5.28}$$

where the weights $w_1, \ldots, w_n$ and samples $X_1, \ldots, X_n$ are those of the kernel mean estimator $\hat{m}_P = \sum_{i=1}^n w_i k_\gamma(\cdot, X_i)$.

This estimator may be seen as a weighted variant of kernel density estimators (Silverman, 1986). In fact, our analysis below requires the bandwidth $h$ to be decreased as the sample size increases, as for kernel density estimation. This estimator

has been used in the literature on kernel mean embeddings (Song and Dai, 2013; Song et al., 2014) without theoretical justifications. Our analysis below provides theoretical guarantees for these works.

We can intuitively explain why the use of the smoothing kernel yields a density estimator as follows. It is well known that the smoothing kernel $J_h(\cdot - x_0)$ converges to the delta function $\delta_{x_0}$ at $x_0 \in \mathcal{X}$ in a certain sense, as the bandwidth $h$ goes to 0. Thus the expectation $\mathbf{E}_{X\sim P}[J_h(X - x_0)]$ of the smoothing kernel converges to the expectation $\mathbf{E}_{X\sim P}[\delta_{x_0}(X)]$ of the delta function, which is the density $p(x_0) = \int \delta_{x_0}(x)p(x)dx$. Note that (5.28) is a consistent estimator of $\mathbf{E}_{X\sim P}[J_h(X - x_0)]$ for a fixed $h$, since $J_h(\cdot - x_0)$ belongs to the Besov space $B^{\alpha}_{2,\infty}(\mathbb{R}^d)$ for any $\alpha > 0$. Therefore (5.28) would be a consistent estimator of the density $p(x_0)$, if $h$ is decreased to 0 with an appropriate speed. This is the reasoning used in our proof. Note that because we take the limit $\alpha \to \infty$ in the proof, the constant $\alpha$ does not appear in the following result.

**Theorem 5.** *Assume that $P$, $Q$, and $\{(w_i, X_i)\}_{i=1}^n$ satisfy the conditions in Theorem 4. Moreover, assume that $P$ has a density function that is bounded and Lipschitz continuous. Let $\xi > 0$ be an arbitrarily small constant, and define $h_n := n^{-\frac{b}{d+1}+\xi}$. Then for any $x_0 \in \mathbb{R}^d$, we have*

$$\mathbf{E}\left[\left|\sum_{i=1}^{n} w_i J_{h_n}(X_i - x_0) - p(x_0)\right|\right] = O\left(n^{-\frac{b}{d+1}+\xi}\right) \quad (n \to \infty). \tag{5.29}$$

Theorem 5 shows that the estimator (5.28) is consistent, and converges to the true density value $p(x_0)$ at a certain rate. This is shown under the assumption that the density function is Lipschitz and bounded. These are standard assumptions used in convergence analysis of kernel density estimation (Silverman, 1986). The schedule of the bandwidth $h_n := n^{-\frac{b}{d+1}}$ is determined so as to balance the terms in an upper bound of the error in (5.29).

The obtained rate $O\left(n^{-\frac{b}{d+1}+\xi}\right)$ depends on the constant $b$, which comes from the rate of the kernel mean estimator $\hat{m}_P$, and on the dimensionality $d$. Note that this happens even if the rate of the kernel mean estimator does not depend on the dimensionality. This matches the fact that any density estimator suffers from the curse of dimensionality.

Consider the case where the samples $X_1, \ldots, X_n$ are i.i.d. with the distribution $P$ and the weights are uniform $w_1 = \cdots = w_n = 1/n$. In this case, the estimator (5) reduces to that of kernel density estimation. The resulting rate of Theorem 5 is $O(n^{-\frac{1}{2+2d}})$, as we have $b = 1/2$ in this case. This is slower than the known min-max optimal rate $O(n^{-\frac{1}{2+d}})$ for the same assumption on the density (Stone, 1980). This suggests that the rate of Theorem 5 might be suboptimal and could be improved.

Note that our density estimator (5) aims at situations different from that of kernel density estimation. Namely, we consider situations where samples from the target distribution $P$ are not available. More precisely, we assume that the kernel mean $m_P$ is estimated by some algorithm (such as the filtering method in Chapter 4), and the density is to be decoded from the resulting kernel mean estimate $\hat{m}_P$. Therefore our result guarantees that the estimator (5) can be used for this purpose.

## 5.5 Numerical experiments

We conduct numerical experiments to verify the theoretical results above. To this end, we consider the following setting.

Let $d = 1$ and $P$ be the normal distribution $\mathbb{N}(0, \sigma_P^2)$ with mean 0 and variance $\sigma_P^2 = 0.01$. We fix the bandwidth of the Gaussian kernel $k_\gamma(x, x') := \exp(-(x - x')^2/2\gamma^2)$ to $\gamma = 0.1$. Letting $P$ be the normal distribution allows us to obtain an analytic expression for the kernel mean $m_P(x) = \sqrt{\frac{\gamma^2}{\sigma^2+\gamma^2}} \exp(-\frac{x^2}{2(\gamma^2+\sigma_P^2)})$. This enables us to evaluate the error $\|m_P - \hat{m}_P\|_{\mathcal{H}_\gamma}$ of a given kernel mean estimate $\hat{m}_P$.

**Kernel mean estimates.** For simplicity, we artificially generated an estimate $\hat{m}_P = \sum_{i=1}^n w_i k_\gamma(\cdot, X_i)$ as follows, based on the true kernel mean $m_P$. First, we generated $n$ samples $X_1, \ldots, X_n$ independently from a uniform distribution on $[-1, 1]$. We then computed the weights $w_1, \ldots, w_n$ by solving the following optimization problem:

$$\min_{w \in \mathbb{R}^n} \| \sum_{i=1}^n w_i k_\gamma(\cdot, X_i) - m_P \|_{\mathcal{H}_\gamma}^2 + \lambda \|w\|^2,$$

where $\lambda > 0$ is a regularization constant. This allows us to control the tradeoff between the error $\|\hat{m}_P - m_P\|_{\mathcal{H}_\gamma}^2$ for the estimate $\hat{m}_P$ and the quantity $\sum_{i=1}^n w_i^2 = \|w\|^2$ defined by the weights. That is, if we let $\lambda$ be very small, then the error $\|\hat{m}_P - m_P\|_{\mathcal{H}_\mathcal{X}}^2$ will be small while the quantity $\sum_{i=1}^n w_i^2$ will be very large, and vice versa. Recall that our theoretical results imply the following: convergence rates of function value expectations may be affected not only by the rate of the kernel mean estimate $\hat{m}_P$, but also by the rate of the quantity $\sum_{i=1}^n w_i^2$ decreasing to 0. Introducing the regularization constant allows us to check whether this is true or not.

**Test functions.** We used the test functions described in Table 5.1 for our experiments. All of these are not included in the Gaussian RKHS $\mathcal{H}_\gamma$. For the function $f(x) = \cos(x)$, we computed the ground-truth value for its the expectation $\mathbf{E}_{X \sim P}[f(X)]$ by numerical integration within precision $1e - 10$.

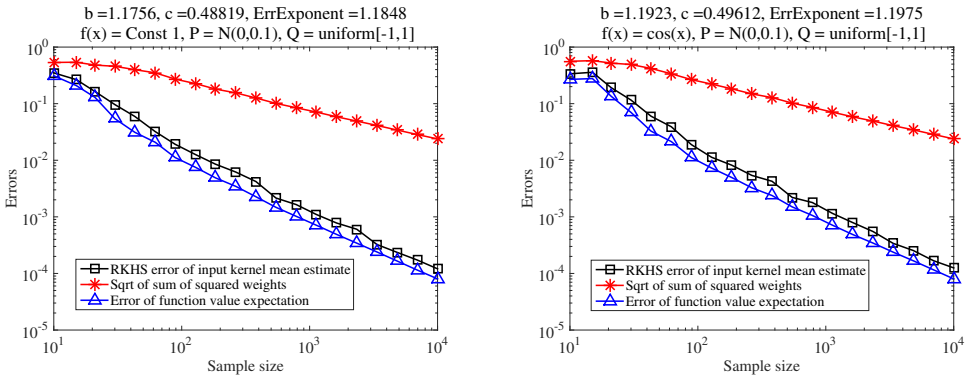
Table 5.1: Test functions

| Function $f(x)$ | Expectation $\mathbf{E}_{X\sim P}[f(X)]$ | Properties |
|---|---|---|
| 1 | 1 | Constant |
| $\cos(x)$ | 0.9950 | Infinitely differentiable |
| $I_{[-0.1,0.1]}(x)$ | 0.6827 | Indicator function |
| $I_{[-\infty,0]}(x)$ | 0.5 | Indicator function |
| $x$ | 0 | Polynomial (mean) |
| $x^2$ | 0.01 | Polynomial (variance) |

For each test function, we conducted the following experiments by varying the sample size $n$ (here we fixed the regularization constant as $\lambda = 0.1$). For each $n$, we generated a kernel mean estimate $\hat{m}_P = \sum_{i=1}^n w_i k_\gamma(\cdot, X_i)$ and measured the absolute error $|\sum_{i=1}^n w_i f(X_i) - \mathbf{E}_{X\sim P}[f(X)]|$: we repeated this 20 times and averaged the results. We also computed the quantity $\sqrt{\sum_{i=1}^n w_i^2}$ in the same way.

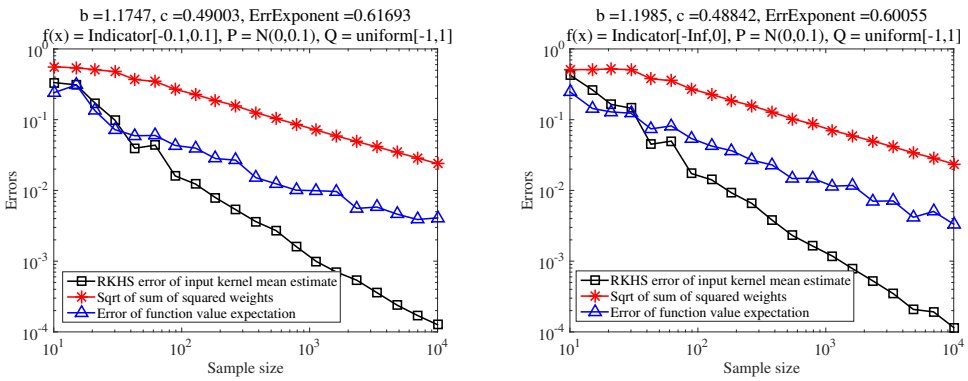
The results are shown in Figures 5.1, 5.2 and 5.3. The black lines are the errors of kernel mean estimates $\|\hat{m}_P - m_P\|_{\mathcal{H}_\gamma}$, the red ones are the values of the quantity $\sqrt{\sum_{i=1}^n w_i^2}$, and the blue ones are the errors of function value estimates $|\sum_{i=1}^n w_i f(X_i) - \mathbf{E}_{X\sim P}[f(X)]|$.

Since the functions $f(x) = 1$ and $f(x) = \cos(x)$ are infinitely differentiable, the errors with respect to these functions decrease in parallel to the errors of kernel mean estimates. This confirms the statement of Proposition 1. For the polynomial functions $f(x) = x$ and $f(x) = x^2$, the rates are even faster than those of kernel mean estimates. On the other hand, as expected, the convergence rates for the indicator functions $f(x) = I_{[-0.1,0.1]}(x)$ and $f(x) = I_{[-\infty,0]}(x)$ are slower than those for the infinitely differentiable functions. This would be because the indicator functions are discontinuous at their boundaries. The slow rates may be explained by Corollary 6, which only guarantees slower rates for the probability mass estimates.



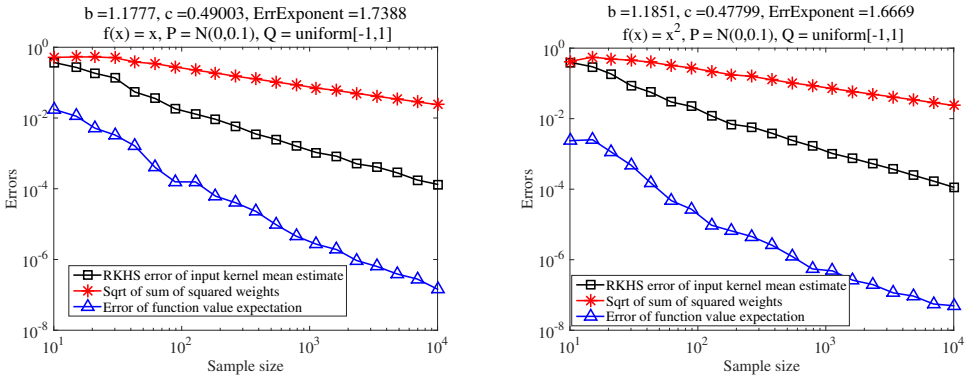

(a) $f(x) = 1$ (b) $f(x) = \cos(x)$

Figure 5.1: Simulation results for function value expectations with infinitely differentiable functions.




(a) $f(x) = I_{[-0.1,0.1]}(x)$ (b) $f(x) = I_{[-\infty,0]}(x)$

Figure 5.2: Simulation results for function value expectations with indicator functions.




(a) $f(x) = x$ (b) $f(x) = x^2$

Figure 5.3: Simulation results for function value expectations with polynomial functions.

## 5.6 Proofs

In the following, $L_p(\nu)$ for arbitrary measure $\nu$ and $p \in (0, \infty]$ denotes the Banach space consisting of $p$-integrable functions with respect to $\nu$. We will use the following inequity in our proofs, which holds for arbitrary $f \in B^{\alpha}_{2,\infty}(\mathcal{X})$:

$$\omega_{r,L_2(\mathcal{X})}(f,t) \leq |f|_{B^{\alpha}_{2,\infty}(\mathcal{X})} t^{\alpha}, \qquad t > 0 \ . \tag{5.30}$$

### 5.6.1 Proof of Theorem 4

Our strategy in the proof of Theorem 4 is to approximate the function in the Besov space by a sequence of functions in the RKHS. A recent study on learning theory has yielded bounds for errors when approximating a Besov function with certain RKHS functions and for their associated RKHS norms (Eberts and Steinwart, 2013, Theorem 2.2., Theorem 2.3). Some of the inequalities derived in our proof use these results.

*Proof.* Let $\gamma_n = n^{-\beta}$, where $\beta > 0$ is a constant determined later. Let $\mathcal{H}_{\gamma_n}$ denote the RKHS of the Gaussian kernel $k_{\gamma_n}$.

First, we show some inequalities needed in the proof. Note that assumption $f \in B^{\alpha}_{2,\infty}(\mathbb{R}^d)$ implies $f \in L_2(\mathbb{R}^d)$. We define $k_\gamma : \mathbb{R}^d \to \mathbb{R}$ by $k_\gamma(x) = \exp(-\|x\|^2/\gamma^2)$ for $\gamma > 0$. Let $r = \lfloor \alpha \rfloor + 1$ and define $K_\gamma : \mathbb{R}^d \to \mathbb{R}$ by

$$K_\gamma(x) := \sum_{j=1}^{r} \binom{r}{j} (-1)^{1-j} \frac{1}{j^d} \left( \frac{2}{\gamma^2 \pi} \right)^{\frac{d}{2}} k_{j\gamma/\sqrt{2}}(x). \tag{5.31}$$

Let $f_n : \mathbb{R}^d \to \mathbb{R}$ be the convolution of $K_{\gamma_n}$ and $f$

$$f_n(x) := (K_{\gamma_n} * f)(x) := \int_{\mathbb{R}^d} K_{\gamma_n}(x-t)f(t)dt \ , \quad x \in \mathbb{R}^d.$$

Then by $f \in L_2(\mathbb{R}^d) \cap L_\infty(\mathbb{R}^d)$, the following inequalities hold by (Eberts and Steinwart, 2013, Theorem 2.2.) and Eq. (5.30):

$$\begin{aligned} & \|f_n - f\|_{L_2(P)} \\ \leq \ & \left(C_{r,1}\|g_1\|_{L_\infty(\mathbb{R}^d)}\right)^{1/2} \omega_{r,L_2(\mathbb{R}^d)}(f, \gamma_n/2) \\ \leq \ & A\gamma_n^\alpha, \end{aligned} \tag{5.32}$$

$$\begin{aligned} & \|f_n - f\|_{L_2(Q)} \\ \leq \ & \left(C_{r,2}\|g_2\|_{L_\infty(\mathbb{R}^d)}\right)^{1/2} \omega_{r,L_2(\mathbb{R}^d)}(f, \gamma_n/2) \\ \leq \ & B\gamma_n^\alpha, \end{aligned} \tag{5.33}$$

where $g_1$ and $g_2$ denote the Lebesgue densities of $P$ and $Q$, respectively, $C_{r,1}$ and $C_{r,2}$ are constants only depending on $r$, and $A$ and $B$ are constants independent of $\gamma_n$.

By $f \in L_2(\mathbb{R}^d)$, (Eberts and Steinwart, 2013, Theorem 2.3.) yields $f_n \in \mathcal{H}_{\gamma_n}$ and

$$\|f_n\|_{\mathcal{H}_{\gamma_n}} \leq C\gamma_n^{-d/2}, \tag{5.34}$$

where $C$ is a constant independent of $\gamma_n$.

We are now ready to prove the assertion. The triangle inequality yields the following inequality:

$$\begin{aligned} & \mathbf{E}\left[\left|\sum_{i=1}^n w_i f(X_i) - \mathbf{E}_{X\sim P}[f(X)]\right|\right] \\ \leq \ & \mathbf{E}\left[\left|\sum_{i=1}^n w_i f(X_i) - \sum_{i=1}^n w_i f_n(X_i)\right|\right] && (5.35) \\ + \ & \mathbf{E}\left[\left|\sum_{i=1}^n w_i f_n(X_i) - \mathbf{E}_{X\sim P}[f_n(X)]\right|\right] && (5.36) \\ + \ & \left|\mathbf{E}_{X\sim P}[f_n(X)] - \mathbf{E}_{X\sim P}[f(X)]\right|. && (5.37) \end{aligned}$$

We first derive a rate of convergence for the first term Eq. (5.35):

$$\begin{aligned}
& \mathbf{E}\left[\left|\sum_{i=1}^{n} w_i f(X_i) - \sum_{i=1}^{n} w_i f_n(X_i)\right|\right] \\
= \;& \mathbf{E}\left[\left|\sum_{i=1}^{n} w_i (f(X_i) - f_n(X_i))\right|\right] \\
\leq \;& \mathbf{E}\left[\left(\sum_{i=1}^{n} w_i^2\right)^{1/2} \left(\sum_{i=1}^{n} (f(X_i) - f_n(X_i))^2\right)^{1/2}\right] \\
\leq \;& \left(\mathbf{E}\left[\sum_{i=1}^{n} w_i^2\right]\right)^{1/2} \\
& \left(\mathbf{E}\left[n\left(\frac{1}{n}\sum_{i=1}^{n} (f(X_i) - f_n(X_i))^2\right)\right]\right)^{1/2} \\
= \;& \left(\mathbf{E}\left[\sum_{i=1}^{n} (w_i)^2\right]\right)^{1/2} n^{1/2} \|f - f_n\|_{L_2(Q)},
\end{aligned}$$

where we used the Cauchy-Schwartz inequality in the first two inequalities. Note that since the weights $w_1, \ldots, w_n$ depend on the random variables $X_1, \ldots, X_n$, the term $(\sum_{i=1}^{n} w_i^2)^{1/2}$ in the third line is not independent of the term $(\sum_{i=1}^{n} (f(X_i) - f_n(X_i))^2)^{1/2}$. By the assumption $\mathbf{E}\left[\sum_{i=1}^{n} (w_i)^2\right] = O(n^{-2c})$, Eq. (5.33), and $\gamma_n = n^{-\beta}$, the rate of the first term is $O\left(n^{-c+1/2-\alpha\beta}\right)$.

We next show a convergence rate for the second term Eq. (5.36):

$$\begin{aligned}
& \mathbf{E}\left[\left|\sum_{i=1}^{n} w_i f_n(X_i) - \mathbf{E}_{X\sim P}[f_n(X)]\right|\right] \\
= \;& \mathbf{E}\left[\langle \hat{m}_P - m_P, f_n \rangle_{\mathcal{H}_{\gamma_n}}\right] \\
\leq \;& \mathbf{E}\left[\|\hat{m}_P - m_P\|_{\mathcal{H}_{\gamma_n}}\right] \|f_n\|_{\mathcal{H}_{\gamma_n}},
\end{aligned}$$

where the equality follows from $f_n \in \mathcal{H}_{\gamma_n}$. By the assumption $\mathbf{E}\left[\|\hat{m}_P - m_P\|_{\mathcal{H}_{\gamma_n}}\right] = O(n^{-b})$, $\gamma_n = n^{-\beta}$, and Eq. (5.34), the rate of the second term is $O\left(n^{-b+\beta d/2}\right)$.

The third term Eq. (5.36) is bounded as

$$\begin{aligned}
|\mathbf{E}_{X\sim P}[f_n(X)] - \mathbf{E}_{X\sim P}[f(X)]| \;&\leq\; \|f_n - f\|_{L_1(P)} \\
&\leq\; \|f_n - f\|_{L_2(P)} \;.
\end{aligned}$$

The rate of this term is $O(n^{-\alpha\beta})$, and this is faster than the first term since $c \leq 1/2$. Thus the overall rate is dominated by the first and second terms.

We chose $\beta$ by balancing the first and second terms, and this yields $\beta = \frac{b-c+1/2}{\alpha+d/2}$. The assertion is derived by substituting this value in the above terms.

□

### 5.6.2 Proof of Proposition 1

*Proof.* We use Stein's extension theorem (Stein, 1970, pp.180-192) (Adams and Fournier, 2003, p.154 and p.230). Let $\mathcal{X} \subset \mathbb{R}^d$ be a set with *minimally smooth boundary* (Stein, 1970, p.189). Stein's extension theorem guarantees that for any $f \in B^{\alpha}_{2,\infty}(\mathcal{X})$, there exists $\mathfrak{E}(f) \in B^{\alpha}_{2,\infty}(\mathbb{R}^d)$ satisfying $\mathfrak{E}(f)(x) = f(x)$ for all $x \in \mathcal{X}$. Likewise, the theorem guarantees that for any $f \in L_{\infty}(\mathcal{X})$, there exists $\mathfrak{E}(f) \in L_{\infty}(\mathbb{R}^d)$ satisfying the same property. Extended function $\mathfrak{E}(f)$ is defined in a way independent of the function space on $\mathcal{X}$ to which $f$ belongs (Stein, 1970, p.191).

Since $B_R$ has minimally smooth boundary (Stein, 1970, p.189), Stein's extension theorem guarantees that for $f$ satisfying Eq. (5.22), there exists $\mathfrak{E}(f) : \mathbb{R}^d \to \mathbb{R}$ such that $\mathfrak{E}(f) \in L_{\infty}(\mathbb{R}^d) \cap B^{\alpha}_{2,\infty}(\mathbb{R}^d)$ and $\mathfrak{E}(f)(x) = f(x), \quad \forall x \in B_R$. Then, applying Theorem 4 to $\mathfrak{E}(f)$, we obtain the rate (5.21) for $\mathbf{E}[|\sum_{i=1}^n w_i \mathfrak{E}(f)(X_i) - E_P[\mathfrak{E}(f)(X)]|]$. Since $B_R$ contains the supports of $P$ and $Q$, we have $\mathbf{E}[|\sum_{i=1}^n w_i \mathfrak{E}(f)(X_i) - E_P[\mathfrak{E}(f)(X)]|] = \mathbf{E}[|\sum_{i=1}^n w_i f(X_i) - E_P[f(X)]|]$. Thus, it turns out that the obtained rate is for $\mathbf{E}[|\sum_{i=1}^n w_i f(X_i) - E_P[f(X)]|]$.

Note that $f$ satisfies Eq. (5.22) for arbitrarily large $\alpha > 0$ since it is infinitely continuously differentiable. Thus, Theorem 4 combined with the above arguments prove the assertion. □

### 5.6.3 Proof of Corollary 6

*Proof.* Let $\mathcal{F}(I_\Omega)$ denote the Fourier transform of $I_\Omega$. It can be easily shown that the function $(1 + \|\cdot\|^2)^{\alpha/2}\mathcal{F}(I_\Omega)(\cdot)$ belongs to $L_2(\mathbb{R}^d)$ for any $\alpha$ satisfying $0 < \alpha < 1/2$. Therefore $I_\Omega$ is included in the fractional order Sobolev space $W_2^{\alpha}(\mathbb{R}^d)$ (Adams and Fournier, 2003, p.252). Since $W_2^{\alpha}(\mathbb{R}^d) \subset B^{\alpha}_{2,\infty}(\mathbb{R}^d)$ holds (Edmunds and Triebel, 1996, pp.26-27, p.44), we have $I_\Omega \in B^{\alpha}_{2,\infty}(\mathbb{R}^d)$.

For an arbitrary constant $\alpha$ satisfying $0 < \alpha < 1/2$ and $2\alpha b - d(1/2 - c) > 0$, Theorem 4 then yields the rate of $O\left(n^{-\frac{2\alpha b - d(1/2-c)}{2\alpha+d}}\right)$ for the lhs of the assertion. Let $\alpha = 1/2 - \zeta$, where $0 < \zeta < 1/2$. Then by the assumption $b - d(1/2 - c) > 0$ we have $2\alpha b - d(1/2 - c) > 0$ for sufficiently small $\zeta$. It is not hard to check that $\frac{2\alpha b - d(1/2-c)}{2\alpha+d}$ is monotonically decreasing as a function of $\zeta$. Therefore in the limit of $\zeta \to 0$ we have the supremum value $\frac{b-d(1/2-c)}{1+d}$ over $\zeta \in (0, 1/2)$. Since we can take an arbitrarily small value for $\zeta$, the assertion of the corollary follows. □

### 5.6.4 Proof of Theorem 5

First, we need the following lemmas.

**Lemma 2.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a Lipschitz function. Then there exists a constant $M > 0$ such that for all $x_0 \in \mathbb{R}^d$ and $h > 0$ we have*

$$\left| \int J_h(x - x_0) f(x) dx - f(x_0) \right| \leq Mh \ . \tag{5.38}$$

**Lemma 3.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function satisfying $f \in B^\alpha_{2,\infty}(\mathbb{R}^d)$ for some $\alpha > 0$. Then for any $h > 0$, we have*

$$|f(\cdot/h)|_{B^\alpha_{2,\infty}(\mathbb{R}^d)} = h^{-\alpha+d/2} |f|_{B^\alpha_{2,\infty}(\mathbb{R}^d)} \ . \tag{5.39}$$

We are now ready to prove Theorem 5.

*Proof.* Let $\gamma_n = n^{-\beta}$ and $h_n = n^{-\tau}$, where $\beta, \tau > 0$ are constants determined later.

Let $\alpha > 0$ be an arbitrary positive constant. We define $J_{h_n,x_0} := h_n^{-d} J_{1,x_0}(\cdot/h_n)$. Since $J_{1,x_0} \in B^\alpha_{2,\infty}(\mathbb{R}^d)$ holds, we then have by Lemma 3

$$|J_{h_n,x_0}|_{B^\alpha_{2,\infty}(\mathbb{R}^d)} = h_n^{-\alpha-d/2} |J_{1,x_0}|_{B^\alpha_{2,\infty}(\mathbb{R}^d)}. \tag{5.40}$$

Let $r := \lfloor \alpha \rfloor + 1$ and define the function $K_\gamma : \mathbb{R}^d \to \mathbb{R}$ by Eq. (5.31). Then by (Eberts and Steinwart, 2013, Theorem 2.2.) and Eqs. (5.30)(5.40) we have

$$\begin{aligned}
&\|K_{\gamma_n} * J_{h_n,x_0} - J_{h_n,x_0}\|_{L_2(P)} \\
&\leq C_1 \omega_{r,L_2(\mathbb{R}^d)}(J_{h_n,x_0}, \gamma_n/2) \\
&\leq C_1' |J_{h_n,x_0}|_{B^\alpha_{2,\infty}(\mathbb{R}^d)} \gamma_n^\alpha \\
&\leq C_1' |J_{1,x_0}|_{B^\alpha_{2,\infty}(\mathbb{R}^d)} h_n^{-\alpha-d/2} \gamma_n^\alpha \ ,
\end{aligned} \tag{5.41}$$

$$\begin{aligned}
&\|K_{\gamma_n} * J_{h_n,x_0} - J_{h_n,x_0}\|_{L_2(Q)} \\
&\leq C_2 \omega_{r,L_2(\mathbb{R}^d)}(J_{h_n,x_0}, \gamma_n/2) \\
&\leq C_2' |J_{h_n,x_0}|_{B^\alpha_{2,\infty}(\mathbb{R}^d)} \gamma_n^\alpha \\
&\leq C_2' |J_{1,x_0}|_{B^\alpha_{2,\infty}(\mathbb{R}^d)} h_n^{-\alpha-d/2} \gamma_n^\alpha,
\end{aligned} \tag{5.42}$$

where $C_1$, $C_1'$, $C_2$, and $C_2'$ are constants independent of $h_n$ and $\gamma_n$.

By (Eberts and Steinwart, 2013, Theorem 2.3.) and Eq. (5.40), we have $K_{\gamma_n,r} *$

$J_{h_n,x_0} \in \mathcal{H}_{\gamma_n}$ and

$$\begin{aligned} & \|K_{\gamma_n} * J_{h_n,x_0}\|_{\mathcal{H}_{\gamma_n}} \\ \leq\ & C_3 \|J_{h_n,x_0}\|_{L_2(\mathbb{R}^d)} \gamma_n^{-d/2} \\ =\ & C_3 \|J_{1,x_0}\|_{L_2(\mathbb{R}^d)} h_n^{-d/2} \gamma_n^{-d/2}, \end{aligned} \tag{5.43}$$

where $C_3$ is a constant independent of $h_n$ and $\gamma_n$.

Similar arguments with the proof of Theorem 4 yields the following inequality:

$$\begin{aligned} & \mathbf{E}\left[\left|\sum_{i=1}^n w_i J_{h_n}(X_i - x_0) - \mathbf{E}_{X\sim P}[J_{h_n}(X - x_0)]\right|\right] \\ \leq\ & \left(\mathbf{E}\left[\sum_{i=1}^n w_i^2\right]\right)^{1/2} n^{1/2} \\ & \|K_{\gamma_n} * J_{h_n,x_0} - J_{h_n,x_0}\|_{L_2(Q)} \\ +\ & \mathbf{E}\left[\|\hat{m}_P - m_P\|_{\mathcal{H}_{\gamma_n}}\right] \|K_{\gamma_n} * J_{h_n,x_0}\|_{\mathcal{H}_{\gamma_n}} \\ +\ & \|K_{\gamma_n} * J_{h_n,x_0} - J_{h_n,x_0}\|_{L_2(P)}\ . \end{aligned}$$

By Eq. (5.42) and the assumption $\mathbf{E}\left[\sum_{i=1}^n w_i^2\right] = O(n^{-c})$, the rate of the first term is $O(n^{-c+1/2-\alpha\beta+\tau(\alpha+d/2)})$. For the second term, Eq. (5.43) and the assumption $\sup_{\gamma>0} \mathbf{E}\left[\|\hat{m}_P - m_P\|_{\mathcal{H}_\gamma}\right] = O(n^{-b})$ yields the rate of $O(n^{-b+\beta d/2+\tau d/2})$. By Eq. (5.41), the rate of the third term is $O(n^{-\alpha\beta+\tau(\alpha+d/2)})$, which is faster than that of the first term.

We chose $\beta$ by balancing the first and second terms, and this yields $\beta = \frac{b-c+1/2+\alpha\tau}{\alpha+d/2}$. By substituting this into the above terms, the overall rate becomes

$$O\left(n^{-\frac{2\alpha(b-\tau d)-d(1/2-c)-d^2\tau/2}{2\alpha+d}}\right)\ . \tag{5.44}$$

Since $\alpha$ can be arbitrarily large, we have for arbitrarily small $\zeta > 0$

$$\begin{aligned} & \mathbf{E}\left[\left|\sum_{i=1}^n w_i J_{h_n}(X_i - x_0) - \mathbf{E}_{X\sim P}[J_{h_n}(X - x_0)]\right|\right] \\ & = O\left(n^{-b+\tau d+\zeta}\right)\ . \end{aligned} \tag{5.45}$$

On the other hand, since

$$\mathbf{E}_{X\sim P}[J_{h_n}(X - x_0)] \ = \ \int J_{h_n}(x - x_0) p(x) dx,$$

Lemma 2 and the Lipschitz continuity of $p$ yield

$$|\mathbf{E}_{X\sim P}[J_{h_n}(X - x_0)] - p(x_0)| \;\leq\; Mh_n = Mn^{-\tau} \;. \tag{5.46}$$

We chose $\tau$ by balancing Eqs. (5.45) and (5.46), and obtain $\tau = \frac{b}{d+1} - \frac{\zeta}{d+1}$. We therefore have $E\left[|\sum_{i=1}^{n} w_i J_{h_n}(X_i - x_0) - p(x_0)|\right] = O\left(n^{-\frac{b}{d+1}+\frac{\zeta}{d+1}}\right)$, and letting $\xi := \frac{\zeta}{d+1}$ yields the assertion of the theorem. □

# Chapter 6

# Conclusions and future work

In this thesis, we have investigated kernel mean embeddings of distributions from the viewpoint of empirical distributions. We have revealed that Monte Carlo methods may be applied to empirical kernel means, as if they were empirical distributions. Based on this theoretical result, we have developed a novel filtering algorithm for state-space models, named Kernel Monte Carlo Filter. We have also conducted theoretical analysis of empirical kernel means: we proved that expectations of functions outside the RKHS can be estimated, when the kernel is Gaussian.

The following are important topics for future work.

**Combinations with other Monte Carlo methods.** While this thesis has provided a framework for combing kernel mean embeddings with particle methods, there are still other possibilities: for example, it would be interesting to consider a combination with MCMC methods.

**Speed up of the resampling algorithm.** While we have discussed how to reduce the computational cost of the resampling algorithm, it is not satisfactory: the reduced cost is still a quadratic order of the sample size. Further speed up may be possible by employing acceleration methods for the Frank-Wolfe algorithm, as this method subsumes Kernel Herding as a special case.

**Parameter estimation with Kernel Monte Carlo Filter.** One interesting direction for extending Kernel Monte Carlo Filter would be parameter estimation for the transition model. In this thesis we did not discuss this, and assumed that parameters are given and fixed, if they exist. If the state observation examples $\{(X_i, Y_i)\}_{i=1}^n$ are given as a sequence from the state-space model, then we can use the state samples $X_1, \ldots, X_n$ for estimating those parameters. Otherwise, we need to estimate the

parameters based on test data. This might be possible by exploiting approaches for parameter estimation in particle methods (e.g., Section IV in Cappé et al. (2007)).

**Transfer learning setting.** Another important topic for the filtering problem is on the situation where the observation model in the test and training phases are different. As discussed in Section 4.2.3, this might be addressed by exploiting the framework of transfer learning (Pan and Yang, 2010). This would require extension of kernel mean embeddings to the setting of transfer learning, since there has been no work in this direction. We consider that such extension is interesting in its own right.

**Kernels other than Gaussian for function value expectations.** Regarding the theory of Chapter 5, it would be desirable to extend the obtained results to kernels other than Gaussian. This would be possible by using the approximation theory based on interpolation spaces or fractional powers of integral operators (Smale and Zhou, 2007).

# References


Adams, R. A. and Fournier, J. J. F. (2003). *Sobolev Spaces*. Academic Press, New York, 2nd edition.

Akaho, S. (2001). A kernel method for canonical correlation analysis. In *Proceedings of the International Meeting on Psychometric Society (IMPS2001)*. Springer-Verlag.

Anderson, B. and Moore, J. (1979). *Optimal Filtering*. Prentice Hall, Englewood Cliffs.

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

Bach, F. and Jordan, M. I. (2002). Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48.

Bach, F., Lacoste-Julien, S., and Obozinski, G. (2012). On the equivalence between herding and conditional gradient algorithms. In *Proceedings of the 29th International Conference on Machine Learning (ICML2012)*, pages 1359–1366.

Berlinet, A. and Thomas-Agnan, C. (2004). *Reproducing kernel Hilbert spaces in probability and statistics*. Kluwer Academic Publisher.

Boots, B., Byravan, A., and Fox, D. (2014). Learning predictive models of a depth camera and manipulator from raw execution traces. In *Proceedings of the 2014 IEEE Conference on Robotics and Automation (ICRA-2014)*.

Boots, B., Gretton, A., and Gordon., G. J. (2013). Hilbert space embeddings of predictive state representations. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI2013)*.

Cappé, O., Godsill, S. J., and Moulines, E. (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *IEEE Proceedings*, 95(5):899–924.

Chen, Y., Welling, M., and Smola, A. (2010). Supersamples from kernel-herding. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI 2010)*, pages 109–116.

DeVore, R. A. and Lorentz, G. G. (1993). *Constructive approximation*. Springer-Verlag, Berlin.

Dick, J., Kuo, F. Y., and Sloan, I. H. (2013). High dimensional numerical integration - the quasi-monte carlo way. *Acta Numerica*, 22(133-288).

Douc, R. and Moulines, E. (2008). Limit theorems for weighted samples with applications to sequential monte carlo methods. *Annals of Statistics*, 36(5):2344–2376.

Doucet, A., Freitas, N. D., and Gordon, N. J., editors (2001). *Sequential Monte Carlo Methods in Practice*. Springer.

Doucet, A. and Johansen, A. M. (2011). A tutorial on particle filtering and smoothing: Fifteen years later. In Crisan, D. and Rozovskii, B., editors, *The Oxford Handbook of Nonlinear Filtering*, pages 656–704. Oxford University Press.

Dudley, R. M. (2002). *Real Analysis and Probability*. Cambridge University Press.

Durbin, J. and Koopman, S. J. (2012). *Time Series Analysis by State Space Methods Second Edition*. Oxford University Press.

Eberts, M. and Steinwart, I. (2013). Optimal regression rates for SVMs using Gaussian kernels. *Electronic Journal of Statistics*, 7:1–42.

Edmunds, D. E. and Triebel, H. (1996). *Function Spaces, Entropy Numbers, Differential Operators*. Cambridge University Press, Cambridge.

Ferris, B., Hähnel, D., and Fox, D. (2006). Gaussian processes for signal strength-based location estimation. In *Proceedings of Robotics: Science and Systems*.

Feuerverger, A. and Mureika, R. A. (1977). The empirical characteristic function and its applications. *Annals of Statistics*, 5(1):88–98.

Fine, S. and Scheinberg, K. (2001). Efficient SVM training using low-rank kernel representations. *Jounal of Machine Learning Research*, 2:243–264.

Flaxman, S., Wang, Y., and Smola, A. (2015). Who supported obama in 2012? ecological inference through distribution regression. In *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD2015)*.

Freund, R. M. and Grigas, P. (2014). New analysis and results for the Frank–Wolfe method. *Mathematical Programming*, DOI 10.1007/s10107-014-0841-6.

Fukumizu, K., Bach, F., and Jordan, M. I. (2004). Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Jounal of Machine Learning Research*, 5:73–99.

Fukumizu, K., Bach, F., and Jordan, M. I. (2009a). Kernel dimension reduction in regression. *The Annals of Statistics*, 37(4):1871–1905.

Fukumizu, K., Gretton, A., Sun, X., and Schölkopf, B. (2008). Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems 20*, pages 489–496.

Fukumizu, K., Song, L., and Gretton, A. (2011). Kernel Bayes' rule. In *Advances in Neural Information Processing Systems 24*, pages 1737–1745.

Fukumizu, K., Song, L., and Gretton, A. (2013). Kernel Bayes' rule: Bayesian inference with positive definite kernels. *Journal of Machine Learning Research*, 14:3753–3783.

Fukumizu, K., Sriperumbudur, B., Gretton, A., and Scholkopf, B. (2009b). Characteristic kernels on groups and semigroups. In *Advances in Neural Information Processing Systems 21*, pages 473–480. MIT Press.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE-Proceedings-F*, 140:107–113.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *Jounal of Machine Learning Research*, 13:723–773.

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. (2005). Measuring statistical dependence with hilbert-schmidt norms. In Jain, S., Simon, H. U., and Tomita, E., editors, *Algorithmic Learning Theory*, volume 3734 of *Lecture Notes in Computer Science*, pages 63–77, Berlin/Heidelberg. Springer-Verlag.

Gretton, A., Fukumizu, K., Teo, C., Song, L., Schoelkopf, B., and Smola, A. (2008). A kernel statistical test of independence. In *Advances in Neural Information Processing Systems 20*, pages 585–592.

Grünewälder, S., Lever, G., Baldassarre, L., Patterson, S., Gretton, A., and Pontil, M. (2012a). Conditional mean embeddings as regressors. In *Proceedings of the 29th International Conference on Machine Learning (ICML2012)*, pages 1823–1830.

Grünewälder, S., Lever, G., Baldassarre, L., Pontil, M., and Gretton, A. (2012b). Modeling transition dynamics in MDPs with RKHS embeddings. In *Proceedings of the 29th International Conference on Machine Learning (ICML2012)*, pages 1823–1830.

Hofmann, T., Schölkopf, B., and Smola, A. J. (2008). Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171–1220.

Jaggi, M. (2013). Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30 th International Conference on Machine Learning*, pages 427–435.

Jitkrittum, W., Gretton, A., Heess, N., Eslami, S., Lakshminarayanan, B., Sejdinovic, D., and Szabo, Z. (2015). Kernel-based just-in-time learning for passing expectation propagation messages. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI2015)*.

Julier, S. J. and Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. In *Proceedings of AeroSense: The 11th International Symposium Aerospace/Defence Sensing, Simulation and Controls*.

Julier, S. J. and Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *IEEE Review*, 92:401–422.

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. and Fukumizu, K. (2014). Recovering distributions from Gaussian RKHS embeddings. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS 2014)*, pages 457–465.

Kanagawa, M., Nishiyama, Y., Gretton, A., and Fukumizu, K. (2014). Monte Carlo filtering using kernel embedding of distributions. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI-14)*, pages 1897–1903.

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *Proceedings of 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag, New York.

McCalman, L., O'Callaghan, S., and Ramos, F. (2013). Multi-modal estimation with kernel embeddings for learning motion models. In *Proceedings of 2013 IEEE International Conference on Robotics and Automation*, pages 2845–2852.

Minh, H. Q. (2010). Some properties of Gaussian reproducing kernel Hilbert spaces and their implications for function approximation and learning theory. *Constructive Approximation*, 32(2):307–338.

Muandet, K., Fukumizu, K., and F. Dinuzzo, B. S. (2012). Learning from distributions via support measure machines. In *Advances in Neural Information Processing Systems 25 (NIPS2012)*, pages 10–18.

Muandet, K. and Schölkopf, B. (2013). One-class support measure machines for group anomaly detection. In *Proceeding of the 29th Conference on Uncertainty in Artificial Intelligence (UAI 2013)*.

Nishiyama, Y., Boularias, A., Gretton, A., and Fukumizu, K. (2012). Hilbert space embeddings of POMDPs. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI2012)*, pages 644–653.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Pistohl, T., Ball, T., Schulze-Bonhage, A., Aertsen, A., and Mehring, C. (2008). Prediction of arm movement trajectories from ECoG-recordings in humans. *Journal of Neuroscience Methods*, 167(1):105–114.

Pronobis, A. and Caputo, B. (2009). COLD: COsy Localization Database. *The International Journal of Robotics Research (IJRR)*, 28(5):588–594.

Quigley, M., Stavens, D., Coates, A., and Thrun, S. (2010). Sub-meter indoor localization in unmodified environments with inexpensive sensors. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2010 (IROS10)*, volume 1, pages 2039–2046.

Ristic, B., Arulampalam, S., and Gordon, N. (2004). *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House.

Robert, C. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer.

Schalk, G., Kubanek, J., Miller, K. J., Anderson, N. R., Leuthardt, E. C., Ojemann, J. G., Limbrick, D., Moran, D., Gerhardt, L. A., and Wolpaw, J. R. (2007). Decoding two-dimensional movement trajectories using electrocorticographic signals in humans. *Journal of Neural Engineering*, 4(264):264–75.

Schölkopf, B., Smola, A., and Müller, K. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels*. MIT Press.

Sejdinovic, D., Gretton, A., and Bergsma, W. (2013a). A kernel test for three-variable interactions. In *Advances in Neural Information Processing Systems 26*.

Sejdinovic, D., Sriperumbudur, B., Gretton, A., and Fukumizu, K. (2013b). Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *Annals of Statistics*, 41(5):2263–2702.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman and Hall.

Smale, S. and Zhou, D. (2007). Learning theory estimates via integral operators and their approximations. *Constructive Approximation*, 26:153–172.

Smola, A., Gretton, A., Song, L., and Schölkopf, B. (2007). A Hilbert space embedding for distributions. In *Proceedings of the International Conference on Algorithmic Learning Theory*, volume 4754, pages 13–31. Springer.

Song, L., Anamdakumar, A., Dai, B., and Xie, B. (2014). Nonparametric estimation of multi-view latent variable models. In *Proceedings of the 31st International Conference on Machine Learning (ICML2014)*, volume 640-648.

Song, L., Boots, B., Siddiqi, S., Gordon, G., and Smola, A. (2010a). Hilbert space embeddings of hidden Markov models. In *Proceedings of the 27th International Conference on Machine Learning (ICML2010)*, pages 991–998.

Song, L. and Dai, B. (2013). Robust low rank kernel embeddings of multivariate distributions. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 26*, pages 3228–3236. Curran Associates, Inc.

Song, L., Fukumizu, K., and Gretton, A. (2013). Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine*, 30(4):98–111.

Song, L., Gretton, A., Bickson, D., Low, Y., and Guestrin, C. (2011a). Kernel belief propagation. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS2011)*, pages 707–715.

Song, L., Gretton, A., and Guestrin, C. (2010b). Nonparametric tree graphical models via kernel embeddings. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS2010)*, pages 765–772.

Song, L., Huang, J., Smola, A., and Fukumizu, K. (2009). Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *Proceedings of the 26th International Conference on Machine Learning (ICML2009)*, pages 961–968.

Song, L., Parikh, A. P., and Xing, E. P. (2011b). Kernel embeddings of latent tree graphical models. In *Advances in Neural Information Processing Systems 25*.

Song, L., Zhang, X., Smola, A., Gretton, A., and Schölkopf, B. (2008). Tailoring density estimation via reproducing kernel moment matching. In *Proceedings of the 25th International Conference on Machine Learning (ICML2008)*, pages 992–999.

Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Schölkopf, B., and Lanckriet, G. R. (2010). Hilbert space embeddings and metrics on probability measures. *Jounal of Machine Learning Research*, 11:1517–1561.

Stein, E. M. (1970). *Singular integrals and differentiability properties of functions*. Princeton University Press, Princeton, NJ.

Steinwart, I. and Christmann, A. (2008). *Support Vector Machines*. Springer.

Stone, C. J. (1977). Consistent nonparametric regression. *The Annals of Statistics*, 5(4):595–620.

Stone, C. J. (1980). Optimal rates of convergence for nonparametric estimators. *The Annals of Statistics*, 8(6):1348–1360.

Szabó, Z., Gretton, A., Póczos, B., and Sriperumbudur, B. K. (2015). Two-stage sampled learning theory on distributions. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS2015)*, pages 948–957.

Székely, G. J. and Rizzo, M. L. (2013). Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143:1249–1272.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press.

van Hoof, H., Peters, J., and Neumann, G. (2015). Learning of non-parametric control policies with high-dimensional state features. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 995–1003.

Vapnik, V. N. (1998). *Statistical Learning Theory*. John Wiley & Sons.

Vlassis, N., Terwijn, B., and Kröse, B. (2002). Auxiliary particle filter robot localization from high-dimensional sensor observations. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 7–12.

Wang, Z., Ji, Q., Miller, K. J., and Schalk, G. (2011). Prior knowledge improves decoding of finger flexion from electrocorticographic signals. *Frontiers in Neuroscience*, 5:127.

Wendland, H. (2005). *Scattered Data Approximation*. Cambridge University Press, Cambridge, UK.

Widom, H. (1963). Asymptotic behavior of the eigenvalues of certain integral equations. *Transactions of the American Mathematical Society*, 109:278–295.

Widom, H. (1964). Asymptotic behavior of the eigenvalues of certain integral equations ii. *Archive for Rational Mechanics and Analysis*, 17:215–229.

Wolf, J., Burgard, W., and Burkhardt, H. (2005). Robust vision-based localization by combining an image retrieval system with monte carlo localization. *IEEE Transactions on Robotics*, 21(2):208–216.

Yoshikawa, Y., Iwata, T., and Sawada, H. (2014). Latent support measure machines for bag-of-words data classification. In *Advances in Neural Information Processing Systems 27*.

Yu, J. (2004). Empirical characteristic function estimation and its applications. *Econometric Reviews*, 23(2):93–123.

Zhu, P., Chen, B., and Príncipe, J. C. (2014). Learning nonlinear generative models of time series with a Kalman filter in RKHS. *IEEE Transactions on Signal Processing*, 62(1):141–155.