Kernel Methods for Monte Carlo

Heiko Strathmann

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Gatsby Unit for Computational Neuroscience and Machine Learning
University College London

December 25, 2017

I, Heiko Strathmann, confirm that the work presented in this thesis, except as noted herein, is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

This thesis investigates the use of reproducing kernel Hilbert spaces (RKHS) in the context of Monte Carlo algorithms. The work proceeds in three main themes.

Adaptive Monte Carlo proposals

We introduce and study two adaptive Markov chain Monte Carlo (MCMC) algorithms to sample from target distributions with non-linear support and intractable gradients. Our algorithms, generalisations of random walk Metropolis and Hamiltonian Monte Carlo, adaptively learn local covariance and gradient structure respectively, by modelling past samples in an RKHS. We further show how to embed these methods into the sequential Monte Carlo framework.

Efficient and principled score estimation

We propose methods for fitting an RKHS exponential family model that work by fitting the gradient of the log density, the *score*, thus avoiding the need to compute a normalization constant. While the problem is of general interest, here we focus on its embedding into the adaptive MCMC context from above. We improve the computational efficiency of an earlier solution with two novel fast approximation schemes without guarantees, and a low-rank, Nyström-like solution. The latter retains the consistency and convergence rates of the exact solution, at lower computational cost.

Goodness-of-fit testing

We propose a non-parametric statistical test for goodness-of-fit. The measure is a divergence constructed via Stein's method using functions from an RKHS. We derive a statistical test, both for i.i.d. and non-i.i.d. samples, and apply the test to quantifying convergence of approximate MCMC methods, statistical model criticism, and evaluating accuracy in non-parametric score estimation.

Acknowledgements

I would like to express my deepest gratitude to my family, my supervisor Arthur Gretton, and the Gatsby unit.

Contents

Chapter 1

Introduction

1.1 Overview

This thesis explores how kernel methods can be used within the wider context of Monte Carlo, a method to solve integration problems, such as Bayesian expectations, using random samples. More specifically, we are concerned with the problems of efficient random sampling, learning structure of unknown densities from data, and measuring how well a set of samples (e.g. obtained from a random sampling algorithm) fits a particular model.

The efficiency of Monte Carlo sampling algorithms, such as Markov chain Monte Carlo (MCMC), crucially relies on the proposal mechanisms that guide how the next step in the Markov chain trajectory is generated. We show how to construct a number of such proposal mechanisms via using kernel methods to model the structure of the underlying density, e.g. covariance or gradients. This results in a flexible framework that can be embedded into MCMC and sequential Monte Carlo algorithms.

The main ingredient for the above ideas is non-parametric modelling of the structure of an unknown probability density. Kernel methods provide an elegant and computationally efficient framework for such modelling tasks, however, in practice they often come with considerable computational costs. In order to ensure practicality of such models in the

Monte Carlo context, we develop novel approximation schemes for existing kernel-based density models. Crucially, we develop statistical guarantees that trade-off introduced error with reduced computational cost.

Finally, the accuracy of Monte Carlo methods strongly depends on the 'quality' of the generated samples, often in the form of bias-variance trade-offs. This is for example the case in the approximate MCMC framework, where the variance of an estimator is reduced at the cost of introducing a (usually small) systematic error. We study how to measure sample quality in this context and develop a non-parametric goodness-of-fit test for non i.i.d. data, which can be used to tune such trade-offs systematically.

1.2 Publications

The thesis structure is based on the following publications. Each chapter is based on collaborative work with the co-authors as outlined in the chapters' beginnings.

Adaptive Monte Carlo proposals

- D. Sejdinovic, H. Strathmann, M. Lomeli, C. Andrieu, and A. Gretton. "Kernel Adaptive Metropolis-Hastings". In: *International Conference for Machine Learning*. 2012

- H. Strathmann, D. Sejdinovic, S. Livingstone, Z. Szabo, and A. Gretton. "Gradient-free Hamiltonian Monte Carlo with Efficient Kernel Exponential Families". In: *Advances in Neural Information Processing Systems*. 2015

- I. Schuster, H. Strathmann, B. Paige, and D. Sejdinovic. "Kernel Adaptive Sequential Monte Carlo". In: *European conference on machine learning & principles and practice of knowledge discovery in databases*. Joint first two authors. 2017

Efficient and principled score estimation

- D. Sutherland, H. Strathmann, M. Arbel, and A. Gretton. "Efficient and principled score estimation". In: *arXiv preprint arXiv:1705.08360* (2017). Joint first two authors. Submitted.

Goodness-of-fit testing

- K. Chwialkowski, H. Strathmann, and A. Gretton. "A kernel test of goodness of fit". In: *International Conference for Machine Learning*. 2016

1.3 Motivation, contribution, and related work

This section provides a brief introduction to the three main themes of this thesis, motivates the research objectives, and points out related work.

1.3.1 Adaptive Monte Carlo proposals

Estimating expectations using Markov Chain Monte Carlo is a fundamental approximate inference technique in Bayesian statistics. Simply speaking, MCMC is a strategy for generating a *Markov chain*, a sequence of random samples $X_1, X_2, \ldots$ from potentially complex and high-dimensional distributions π in a space $\mathcal{X}$, where X_{t+1} depends only on X_t. Using those samples, expectations can be estimated as

$$\int_{\mathcal{X}} f(x)\pi(x)\mathrm{d}x \approx \frac{1}{n}\sum_{i=1}^{n} f(X_i), \tag{1.1}$$

given certain properties of the X_i, and π-integrable f, [6]. For example $f(x) = x$ yields the mean of π.

The key building-block of the MCMC algorithms that we consider here is the Metropolis-Hastings algorithm, an algorithm that given a state X_t produces a proposal for X_{t+1}, and accepts or rejects it at random with a probability that is based on the underlying density π. Since the expected estimation error in (1.1) directly depends on the correlation between successive points in the Markov chain [93], efficiency can be achieved by taking steps with little correlation (i.e. large) that are accepted with high probability. Proposal mechanisms that adapt to the target on the fly aim to strike this balance. Care has to be taken though – using the Markov chain history to inform future moves generally compromises convergence properties of the resulting MCMC algorithm [6].

We will generally assume that the target density π is only available point-wise (in closed form or via a random estimator without bias), and no higher order information, such as gradients, is available. For example, in pseudo-marginal MCMC [5, 16], the target density does not have an analytically tractable expression, but can only be estimated at any given point, e.g. Bayesian Gaussian process classification [44]. A related setting is MCMC for approximate Bayesian computation (ABC), where a Bayesian posterior is approximated through repeated simulation from its likelihood [80, 108]. In those cases, sophisticated gradient-based schemes [50, 95] cannot be applied directly on the intractable target.

Kernel adaptive Metropolis-Hastings

The choice of the proposal distribution is known to be crucial for the design of Metropolis-Hastings algorithms, and methods for adapting the proposal distribution to increase the sampler's efficiency based on the history of the Markov chain have been widely studied [6, 60, 61]. These methods often aim to learn the global covariance structure of the target distribution, and adapt the proposal accordingly.

Contribution. We develop a kernel adaptive Metropolis-Hastings algorithm (KAMH) in which the Markov chain trajectory is mapped to an RKHS, and the proposal distribution is chosen according to the covariance in this feature space [8, 110]. Unlike earlier adaptive approaches, the resulting proposals are locally adaptive in the input space, and oriented towards nearby regions of high density, rather than simply matching the global covariance structure of the distribution. Our approach combines a move in the feature space with a stochastic step towards the nearest input space point, where the feature space move can be analytically integrated out. Thus, the implementation of the procedure is straightforward: the proposal is simply a Gaussian in the input space, with location-dependent covariance that is informed by the feature space representation of the target. Furthermore, the resulting sampler only requires the ability to evaluate the un-normalised density of the target.

Related work. Adaptive MCMC samplers were first studied by Haario et al. [60, 61], who proposed to update the proposal along the sampling process. Based on the chain history, they estimate the covariance of the target distribution and construct a Gaussian proposal centred at the current chain state, with a particular choice of the scaling factor from [48]. More sophisticated schemes are presented by Andrieu and Thoms [6], e.g. adaptive scaling, component-wise scaling, and principal component updates. While these strategies are beneficial for distributions that show high anisotropy (i.e. by ensuring the proposal uses the right scaling in all principal directions), they may still suffer from low acceptance probability and slow mixing when the target distributions are strongly non-linear, and the directions of large variance depend on the current location of the sampler in the support.

Kernel Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) [86] is an MCMC algorithm that improves efficiency by exploiting gradient information. It simulates particle movement along the contour lines of a dynamical system constructed from the target density. Projections of these trajectories cover wide parts of the target's support, and the probability of accepting a move along a trajectory is often close to one. Remarkably, this property is very robust to growing dimensionality, and HMC here often is superior to random walk methods, which need to decrease their step size at a much faster rate [86, Section 4.4]. Unfortunately, as discussed earlier, for a large class of problems gradient information is not available. In those cases, HMC cannot be applied, leaving random walk methods as the only mature alternative.

Contribution. We extend the above idea of using kernel methods to learn local covariance structure for efficient proposal distributions. Rather than *locally* smoothing the target density, however, we estimate its gradients *globally*. This leads to a kernel-based gradient-free adaptive MCMC algorithm, *(KMC)*, that starts as a random walk and then smoothly transitions into HMC-like behaviour. In experiments, KMC outperforms random walk based sampling methods, including the earlier kernel adaptive Metropolis-Hastings. As the latter, KMC is does not require access to target gradients, but estimates those from the sample path only.

Related work. Gaussian Processes (GP) were used by Rasmussen [90] as a surrogate of the target density in order to speed up HMC, however, this requires access to the target in closed form, to provide training points for the GP. More recently, neural networks were used for the same purpose [136]. There have been efforts to mimic HMC's behaviour using stochastic gradients from mini-batches in 'Big Data' [31], or stochastic finite differences in approximate Bayesian computation [82, discussed in Section 4.4.4]. Stochastic gradient based HMC methods, however, often suffer from low acceptance rates or additional bias that is hard to quantify [20].

Score estimation for kernel exponential families

The gradient estimator in KMC is based on a recently proposed procedure to fit a *infinite dimensional* exponential family models to sample points drawn i.i.d. from a probability density [117]. While *finite dimensional* exponential families are a keystone of parametric statistics [25, 126], it is difficult to construct a practical, consistent maximum likelihood solution for infinite dimensional natural parameters [15, 47, 57]. Sriperumbudur et al. [117] proposed to employ a score matching procedure [63], which minimizes the *Fisher distance*: the expected squared distance between the model *score* (i.e. the derivative of the log model density) and the score of the (unknown) true density. The Fisher distance can be recast to yield a quadratic loss using integration by parts. Unlike the maximum likelihood case, a solution can be formulated to obtain a well-posed and straightforward solution, which is a linear system defined in terms of the first and second derivatives of the RKHS kernels at the sample points.

We require our adaptive MCMC algorithms to be computationally efficient yet expressive, as they deal with high-dimensional MCMC chains of growing length. For a practical implementation, it is necessary to approximate the full solution from [117], which has quadratic memory costs and cubic computational costs in both number of samples and dimension. Despite being useful for the developed MCMC proposals, efficiently estimating the natural parameter of such an infinite dimensional exponential family model is a challenging problem on its own.

Contribution. We develop two novel approximations for the kernel exponential family model. The first approximation, *score matching lite*, is based on computing the solution in terms of a lower dimensional, yet growing, and simpler subspace in the RKHS. The second approximation uses a finite dimensional feature space (*score matching finite*), combined with random Fourier features [89]. These approximations greatly reduce computational costs, and can be used on their own or within the gradient-free HMC framework described above.

Related work. This setting of score estimation is closely related to that of *energy-based learning* [72]. For example, Alain and Bengio [2] proposed a deep learning-based approach to directly learn a score function from samples: de-noising auto-encoders are networks trained to recover the original inputs from noise-corrupted versions. We come back to this approach in Chapter 6 where we describe the method in Chapter 6 and compare our methods experimentally in Section 6.3.

Despite the Hamiltonian Monte Carlo context in Chapter 4, the score function is used for constructing control functionals for Monte Carlo integration [88], where learned score functions could be used where closed-form expressions do not exist.

The form of score matching lite is similar to an estimator by Hyvärinen [64], which we will comment on in Section 4.2.1.

Kernel sequential Monte Carlo

In contrast to MCMC, sequential Monte Carlo (SMC) methods are based on iterative importance sampling, and have traditionally been applied to inference in filtering problems with a sequence of time-varying target distributions [39]. We focus on static SMC methods for Bayesian inference on a *fixed* target distribution [29, 32, 43, 105]. Static SMC frames inference as a sequential problem by defining an artificial series of incremental targets. This can be done by tempering the target density [105], by including data points sequentially [32], or by targeting the full density at every iteration. The latter is a special case known as population Monte Carlo (PMC) [28]. SMC offers three striking advantages over MCMC: adaptive proposal mechanisms do not compromise convergence, normalising constants (e.g. model evidence) can be estimated in a straightforward manner, and the particle system can represent multi-modality (where MCMC often gets 'stuck' in a single mode).

Contribution. We develop a framework for kernel sequential Monte Carlo, *(KSMC)*. Similarly to the presented work in adaptive MCMC proposals, KSMC represents the (weighted) particle system of SMC algorithms in an RKHS. The learned structure of the surrogate model is used to construct proposal distributions that are used within SMC. We exemplify this framework with two existing SMC algorithms, again based on covariance and gradients. KSMC enjoys the benefits that SMC has over MCMC, yet the use of kernel surrogates leads to faster convergence for non-linear targets as compared to plain SMC. As before, KSMC does not require target gradient information and therefore is particularly useful in combination with importance sampling frameworks that are based around intractable targets [33, 125].

Related work. Our algorithms are kernel-based generalisations of adaptive SMC [43] and gradient importance sampling [102]. We briefly discuss these algorithms in Section 5.1.2. The local adaptive importance sampling approach by Givens and Raftery [51] adapts to local covariance structure of the target, by computing pairwise distances and then only using a small number of close points to estimate local covariance.

1.3.2 Efficient and principled score estimation

Coming back to approximating the kernel exponential family described in Section 1.3.1, we now focus on the question of developing efficient estimators that are *theoretically* well justified. Indeed, one of the desirable properties of the original estimator by Sriperumbudur et al. [117] is established theory on consistency, with error guarantees depending on the smoothness of the data-generating density.

Contribution. While the approximations in Chapter 4 greatly improve the runtime, no convergence guarantees are known, nor any means of determining how quickly to increase the complexity of these solutions with increasing sample size n. We here develop a learning scheme using the framework of Nyström approximation [112, 130]: representing the solution of the score matching optimisation problem in a sub-space of the original solution. We prove guarantees on the convergence of this algorithm for an increasing number m of Nyström basis points. Depending on the problem difficulty, convergence is attained in the regime $m \sim n^{1/3}$ to $m \sim n^{1/2}$, thus yielding cost savings at the same level of guaranteed generalisation error. The overall Fisher distance between our solution and the true density decreases as $m, n \to \infty$ with rates that *match* those of the full solution in [117, Theorem 6]. An experimental evaluation confirms the performance benefits in practice.

Related work. Guarantees on the performance of Nyström methods have been the topic of considerable study. Earlier approaches have worked by first bounding the error in a Nyström approximation of the kernel matrix on the sample [41], and then separately evaluating the impact of regression with an approximate kernel matrix [37]. This approach, however, results in suboptimal rates; better rates can be obtained by considering the whole problem at once [42], including its direct impact on generalisation error [99]. Approaches like those of [42, 134], which bound the difference in training error of Nyström-type approximations to kernel ridge regression, are insufficient for our purposes: we need to ensure that the estimated unnormalised log-density converges to the truth everywhere, so that the full distribution matches, not just its values at the training points. In doing so, our work is heavily indebted to Caponnetto and De Vito [27], as are Rudi et al. [99] and Sriperumbudur et al. [117].

1.3.3 Goodness-of-fit testing

Further downstream the pipeline of sampling algorithms, the effectiveness of any Monte Carlo algorithm strongly depends on the quality of the used random samples. This is in particular true for approximate sampling methods, which use modifications to Markov transition kernels that improve mixing speed at the cost of introducing asymptotic bias [12, 68, 128]. The resulting bias-variance trade-off can usually be tuned with parameters of the sampling algorithms. It is therefore important to test whether for a particular parameter setting and run-time, the samples are of the desired quality. This question cannot be answered with classical MCMC convergence statistics, such as the widely used potential scale reduction factor (R-factor) [49] or the effective sample size, since these assume that the Markov chain reaches the true equilibrium distribution i.e. absence of asymptotic bias.

We address this question using statistical tests of goodness-of-fit, which are a fundamental tool in statistical analysis, dating back to the test of Kolmogorov and Smirnov [67, 109]. Given a set of samples from a distribution q, our interest is in whether q matches some reference or target distribution p, which we assume to be only known up to the normalisation constant.

Recently, in the multivariate setting, Gorham and Mackey [52] proposed an elegant measure of sample quality with respect to a target. This measure is a maximum discrepancy between empirical sample expectations and target expectations over a large class of test functions, constructed so as to have zero expectation over the target distribution by use of a Stein operator. This operator depends only on the derivative of the $\log q$: thus, the approach can be applied very generally, as it does not require closed-form integrals over the target distribution (or numerical approximations of such integrals). This property is particularly useful in assessing MCMC, since these integrals are certainly not known to the practitioner.

An important application of a goodness-of-fit measure is in *statistical testing*, where it is desired to determine whether the empirical discrepancy measure is large enough to reject the null hypothesis (that the sample arises from the target distribution) with a certain p-value. One approach is to establish the asymptotic behaviour of the test statistic, and to set a test threshold at a large quantile of the asymptotic distribution.

Contribution. We define a statistical test of goodness-of-fit, based on a Stein discrepancy computed in an RKHS. To construct our test statistic, we use a function class defined by applying the Stein operator to a chosen space of RKHS functions, as proposed by Oates et al. [88].[1] Our measure of goodness of fit is the largest discrepancy over this space of functions between empirical sample expectations and target expectations (the latter being zero, due to the effect of the Stein operator). The approach is a natural extension to goodness-of-fit testing of the earlier kernel statistical tests [54, 55], which are based on the *maximum mean discrepancy*, c.f. Section 2.1. As with these earlier tests, our statistic is a simple V-statistic, and can be computed in closed form and in quadratic time in the number of samples. Moreover, it is an unbiased estimate of the corresponding population discrepancy. Only the gradient of the log target density is needed; we do not require integrals with respect to the target density – including the normalisation constant, which makes the test well suited for MCMC samples. We make use of the extensive literature on asymptotics of V-statistics to

[1] Oates et al. addressed the problem of variance reduction in Monte Carlo integration, using the Stein operator to avoid bias.

formulate a hypothesis test [74, 106] and provide statistical tests for both uncorrelated and correlated samples, where the latter is essential if the test is used in assessing the quality of output of an MCMC procedure.

Related work. Several alternative approaches exist in the statistics literature to goodness-of-fit testing. A first strategy is to partition the space, and to conduct the test on a histogram estimate of the distribution [14, 17, 58, 59]. Such space partitioning approaches can have attractive theoretical properties (e.g. distribution-free test thresholds) and work well in low dimensions, however they are much less powerful than alternatives once the dimensionality increases [56]. A second popular approach has been to use the smoothed L_2 distance between the empirical characteristic function of the sample, and the characteristic function of the target density. This dates back to the test of Gaussianity of Baringhaus and Henze [13, Equation 2.1], who used an exponentiated quadratic smoothing function. For this choice of smoothing function, their statistic is identical to the maximum mean discrepancy, c.f. Section 2.1, with the exponentiated quadratic kernel, which can be shown using the Bochner representation of the kernel [115, Corollary 4]. It is essential in this case that the target distribution be Gaussian, since the convolution with the kernel (or in the Fourier domain, the smoothing function) must be available in closed form. An L_2 distance between Parzen window estimates can also be used [24], giving the same expression again, although the optimal choice of bandwidth for consistent Parzen window estimates may not be a good choice for testing [3]. A different smoothing scheme in the frequency domain results in an energy distance statistic (this likewise being an MMD with a particular choice of kernel; see Sejdinovic et al. [103]), which can be used in a test of normality [124]. The key point is that the required integrals are again computable in closed form for the Gaussian, although the reasoning may be extended to certain other families of interest, e.g. [91]. The requirement of computing closed-form integrals with respect to the test distribution severely restricts this testing strategy. Finally, a problem related to goodness-of-fit testing is that of model criticism [79]. In this setting, samples generated from a fitted model are compared via the maximum mean discrepancy, c.f. Section 2.1, with samples used to train the model, such that a small MMD indicates a good fit. There are two limitation to the method: first, it requires samples from the model (which might not be easy if this requires a complex MCMC sampler); second, the choice of number of samples from the model is not obvious, since too few samples cause a loss in test power, and too many are computationally wasteful. Neither issue arises in our test, as we do not require model samples.

An identical test (for uncorrelated samples) was independently developed at the same time by Liu et al. [77].

Thesis structure

We begin by introducing a minimal amount of necessary concepts and notations in Chapter 2. This includes basics of reproducing kernel Hilbert spaces in Section 2.1, score matching in Section 2.2, and Markov chain Monte Carlo in Section 2.3.

Part I – Adaptive Monte Carlo proposals

We introduce three contributions for using kernel methods in adaptive MCMC transition kernels. Chapter 3 is based on modelling covariance in an RKHS for proposals that locally align with the target density. Chapter 4 is based on modelling the score to mimic Hamiltonian dynamics and develops fast approximations to existing kernel-based score learning methods. Chapter 5 applies these concepts to the context of sequential Monte Carlo.

Part II – Efficient and principled score estimation

Chapter 6 provides yet another fast approximate kernel-based score learning method, which is a generalisation of those described in Chapter 4. The method's theoretical foundations are explored in a longer technical proof in Chapter 7.

Part III – Goodness-of-fit testing

Chapter 8 introduces a novel kernel goodness-of-fit test for measuring the quality of MCMC samples, and for assessing convergence of the score estimation models from Chapter 4.

Part IV – Conclusions

We finally conclude, and discuss potential future work and recent developments related to this work.

Chapter 2

Background

We now introduce concepts and notation that is used throughout the rest of the thesis.

2.1 Reproducing kernel Hilbert spaces

Reproducing kernel and feature map

According to the Moore-Aronszajn theorem [19, page 19], for every symmetric, positive definite function (*kernel*)

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$

there is an associated RKHS $\mathcal{H}$ of real-valued functions on $\mathcal{X}$ with reproducing kernel k, that is

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

The map $\varphi : \mathcal{X} \to \mathcal{H}$, with

$$\varphi : x \mapsto k(\cdot, x), \tag{2.2}$$

is called the *(canonical) feature map* of k. The reproducing property (2.1) also holds for for kernel derivatives, as long as k is differentiable [119, Lemma

4.34],

$$\left\langle \frac{\partial}{\partial x_i} k(\cdot, x), f \right\rangle_{\mathcal{H}} = \frac{\partial}{\partial x_i} f(x) \qquad \forall f \in \mathcal{H}, \tag{2.3}$$

which can be generalised to higher-order derivatives as well.

Mean embedding

This feature map or embedding of a single point can be extended to that of a probability measure P on $\mathcal{X}$: its kernel embedding is an element $\mu_P \in \mathcal{H}$, given by

$$\mu_P = \int k(\cdot, x)\, dP(x)$$

[19, 46, 111]. If a measurable kernel k is bounded, it is straightforward to show using the Riesz representation theorem that the mean embedding μ_P exists for all probability measures on $\mathcal{X}$. For many interesting bounded kernels k, including the Gaussian, Laplace and inverse multi-quadratics, the kernel embedding $P \mapsto \mu_P$ is injective. Such kernels are said to be *characteristic* [114, 115], since each distribution is uniquely characterized by its embedding. The kernel embedding μ_P is the representer of expectations of smooth functions w.r.t. P, i.e.

$$\langle f, \mu_P \rangle_{\mathcal{H}} = \int f(x) dP(x) \qquad \forall f \in \mathcal{H}.$$

Given samples $X = \{X_i\}_{i=1}^n \sim P$, the embedding of the empirical measure is $\mu_X = \frac{1}{n} \sum_{i=1}^n k(\cdot, X_i)$, i.e. the sample mean in feature space.

Covariance operator

Denote by $C_P : \mathcal{H} \to \mathcal{H}$ the *covariance operator* for a probability measure P [10, 46], which satisfies

$$C_P = \int k(\cdot, x) \otimes k(\cdot, x)\, dP(x) - \mu_P \otimes \mu_P, \tag{2.4}$$

where for $a,b,c \in \mathcal{H}$ the tensor product is defined as

$$(a \otimes b)c = \langle b,c \rangle_{\mathcal{H}} a. \tag{2.5}$$

The covariance operator has the property that

$$\langle f, C_P g \rangle_{\mathcal{H}} = \mathbb{E}_P(fg) - \mathbb{E}_P f \, \mathbb{E}_P g \qquad \forall f,g \in \mathcal{H}.$$

Analogous to the mean in feature space, we can think of the covariance operator as the generalisation of the covariance matrix in feature space.

Maximum mean discrepancy

The MMD [22, 53, 54] is a measure of distance for two random variables $x \sim P, y \sim Q$, here defined in a unit ball in an RKHS $\mathcal{H}$,

$$\sup_{\|f\|_{\mathcal{H}}=1} \left(\mathbb{E}_P f(x) - \mathbb{E}_Q f(y)\right) = \|\mu_P - \mu_Q\|_{\mathcal{H}},$$

where the right hand side is an interpretation as the mean difference of the random variables embedded in the RKHS, c.f. mean embedding. The equality can be showed using simple RKHS arguments [53, Lemma 4]. The MMD can be estimated from samples $\{X_i\}_{i=1}^n, \{Y_j\}_{j=1}^m$ in quadratic time, for example as

$$\frac{1}{n(n-1)} \sum_{i=1}^n \sum_{j \neq i}^m k(X_i, X_j) + \frac{1}{m(m-1)} \sum_{i=1}^n \sum_{j \neq i}^m k(Y_i, Y_j) - \frac{2}{nm} \sum_{i=1}^n \sum_{j=1}^m k(X_i, Y_j).$$

Measuring convergence of mixed third order moments. Throughout the experimental section of the paper, we will use the empirical MMD in the RKHS of the third order polynomial kernel

$$k(x,y) = \left(1 + x^\top y\right)^3.$$

The corresponding (finite dimensional) RKHS captures all mixed moments up to order three, i.e. an empirical estimate quantifies how close two sets of samples are in terms of such moments in a single quantity. We will compare a reference set of samples from a long MCMC chain with outputs of sampling algorithms.

Kernel parametrization of exponential families

We now describe the *kernel* or *infinite dimensional exponential family* [26, 47]. This is a generalisation of the standard exponential family model [126]. The model class is defined as a set of density functions

$$\mathcal{P} = \left\{ p_f(x) := \exp\left(f(x) - A(f)\right) q_0(x) \mid f \in \mathcal{F} \right\}, \tag{2.6}$$

where $\mathcal{F} \subseteq \mathcal{H}$ is the set of functions in the RKHS $\mathcal{H}$ for which the normalizer

$$A(f) = \log Z(p_f) \qquad Z(p_f) = \int \exp(f(x))\, q_0(x) \mathrm{d}x \tag{2.7}$$

is finite, and q_0 is a base measure with appropriately vanishing tails. That this is a member of the exponential family becomes apparent when we recall the reproducing property (2.1): the feature map $x \mapsto k(x, \cdot)$ in (2.2) is the sufficient statistic, and f is the natural parameter whose evaluation by the reproducing property (2.1) is the inner product with the sufficient statistic, $f(x) = \langle k(x, \cdot), f \rangle_{\mathcal{H}}$. An overview of various finite dimensional members of the exponential family (such as Gamma, Poisson, Binomial, etc.) and their corresponding kernel functions may be found in [117, Example 1]. The model (2.6) defines broad class of densities: when universal kernels are used, the family is dense in the space of continuous densities on compact domains, with respect to e.g. total variation distance and Kulback-Leibler divergence [117, Section 3].

Unfortunately, fitting (2.6) by maximum likelihood becomes impractical in high dimensions, and is ill-posed in infinite dimensions, due to the intractability of $A(f)$ [15, 57], [47, Section 1.3.1]. It is, however, possible to consistently fit an *un-normalised* version of (2.6) by directly minimising the expected gradient mismatch between the model (2.6) and the data generating density. This is achieved by generalising a score matching approach [63, 117] to RKHS parameter spaces. The technique avoids the problem of dealing with the intractable $A(f)$ in (2.7), and reduces the problem to solving a linear system. We describe the estimator below.

2.2 Score matching & un-normalised density estimation

Suppose we are given a set of points $X = \{X_b\}_{b\in[n]} \subset \mathbb{R}^d$ sampled i.i.d. from an unknown distribution with density p_0. Our setting is that of *un-normalised density estimation*, where we wish to fit a model p such that $p(x)/Z(p) \approx p_0(x)$ in some divergence measure, and we do not concern ourselves with the normalization factor in (2.7).

Hyvärinen [63] proposed an elegant approach to fit an un-normalised density p_0 with a model p, by minimizing the Fisher divergence, the expected squared distance between *score functions*[1] $\nabla_x \log p(x)$:

$$J(p_0\|p) = \frac{1}{2}\int p_0(x)\,\|\nabla_x \log p(x) - \nabla_x \log p_0(x)\|_2^2 \mathrm{d}x \tag{2.8}$$

$$= \int p_0(x) \sum_{i=1}^{d} \left[\partial_i^2 \log p(x) + \frac{1}{2}\left(\partial_i \log p(x)\right)^2\right] \mathrm{d}x + \text{const}, \tag{2.9}$$

where we use $\partial_i f(x)$ to denote $\frac{\partial}{\partial x_i} f(x)$. The equality in (2.9) assumes some mild regularity conditions and contains a constant depending only on p_0. Crucially, (2.9) is independent of the normaliser (2.7) and, other than the constant, depends on p_0 only through an expectation, so it can be estimated by a simple Monte Carlo average.

[1] Here we use the term *score* in the sense of Hyvärinen [63], i.e. the gradient of the log density w.r.t. a point x in the domain of p_0.

Kernel exponential families

For the kernel exponential family model (2.6), Sriperumbudur et al. [117, Theorem 3] showed that the score matching loss (2.9) takes a quadratic form (see also Lemma 1 in Chapter 7),

$$J(p_0||p_f) = \frac{1}{2}\langle f - f_0, C(f - f_0)\rangle_{\mathcal{H}},$$

where

$$C : \mathcal{H} \to \mathcal{H}, \quad C := \int p_0 \sum_{i=1}^{d} \partial_i k(\cdot, x) \otimes \partial_i k(\cdot, x) \mathrm{d}x$$

can be thought of as a 'covariance between derivatives' operator similar to the covariance operator (2.4), and $f_0 \in \mathcal{H}$ is the RKHS function that corresponds to the true[2] density p_0. The expression can be further simplified to

$$J(p_0\|p_f) = \frac{1}{2}\langle f, Cf\rangle_{\mathcal{H}} + \langle f, \xi\rangle_{\mathcal{H}} + J(p_0\|q_0),$$

where

$$\mathcal{H} \ni \xi := \int p_0 \sum_{i=1}^{d} \Big(\partial_i k(\cdot, x)\partial_i \log q_0 + \partial_i^2 k(\cdot, x)\Big) \mathrm{d}x,$$

and where $J(p_0\|q_0)$ is the Fisher divergence between the base measure q_0 and the true density (independent of f).

We will mostly use $q_0 = \text{const}$ throughout this work, which results, intuitively speaking, in $\langle \xi, f\rangle_{\mathcal{H}} = \int p_0 \sum_{i=1}^{d} \partial_i^2 f(x)\mathrm{d}x$ quantifying curvature of f.

Estimation in kernel exponential families

We briefly discuss the score matching approach that Sriperumbudur et al. [117] propose to fit the kernel exponential family model (2.6), i.e. to find an

[2] Note that this assumes the 'well specified' case, i.e. there exists an $f_0 \in \mathcal{H}$ such that $p_{f_0} = p_0$, more details will follow in Chapter 6 and Chapter 7.

f such that p_f approximates p_0. Their empirical estimator of (2.9) is

$$\begin{aligned}\hat{J}(f) &= \hat{J}(p_0 \| p_f) - \text{const}\\ &= \frac{1}{n}\sum_{b=1}^{n}\sum_{i=1}^{d} \partial_i^2 f(X_b) + \frac{1}{2}\left(\partial_i f(X_b)\right)^2, \end{aligned} \tag{2.10}$$

where the constant depends on p_0 and q_0 but not on f. Minimizing a regularised version of (2.10) gives

$$f_{\lambda,n} = \underset{f\in\mathcal{H}}{\operatorname{argmin}} \hat{J}(f) + \frac{1}{2}\lambda\|f\|_{\mathcal{H}}^2 = -\frac{\hat{\xi}}{\lambda} + \sum_{a=1}^{n}\sum_{i=1}^{d}\beta_{(a,i)}\partial_i k(X_a,\cdot), \tag{2.11}$$

$$\hat{\xi} = \frac{1}{n}\sum_{a=1}^{n}\sum_{i=1}^{d}\partial_i^2 k(X_a,\cdot) + \partial_i k(X_a,\cdot)\partial_i \log q_0(X_a),. \tag{2.12}$$

where $\beta_{(a,i)}$ denotes the $(a-1)d+i$th entry of a vector $\beta \in \mathbb{R}^{nd}$; we use $\partial_i k(x,y)$ to mean $\frac{\partial}{\partial x_i}k(x,y)$, and $\partial_{i+d}k(x,y)$ for $\frac{\partial}{\partial y_i}k(x,y)$. To evaluate the estimated un-normalised log-density $f_{\lambda,n}$ at a point x, we take a linear combination of $\partial_i k(X_a,x)$ and $\partial_i^2 k(X_a,x)$ for each sample X_a. The weights β of this linear combination are obtained by solving the nd-dimensional linear system

$$(G+n\lambda I)\beta = h/\lambda, \tag{2.13}$$

where $G \in \mathbb{R}^{nd\times nd}$ is the matrix collecting partial derivatives of the kernel at the training points,

$$G_{(a,i),(b,j)} = \partial_i\partial_{j+d}k(X_a,X_b),$$

and $h \in \mathbb{R}^{nd}$ evaluates derivatives of $\hat{\xi}$,

$$h_{(b,i)} = \partial_i\hat{\xi}(X_b).$$

Solving (2.13) takes $\mathcal{O}(n^3d^3)$ time and $\mathcal{O}(n^2d^2)$ memory, which quickly becomes infeasible as n grows, especially for large d.

2.3 Markov chain Monte Carlo

Denote an un-normalised *target* density on $\mathcal{X}$ by π. We are interested in constructing a Markov chain $X_1 \to X_2 \to \cdots \to X_t \to \ldots$ such that $\lim_{t\to\infty} X_t \sim \pi$. By running the Markov chain for a long time T, we can consistently approximate any expectation w.r.t. π using sample averages using (1.1), [93]. Markov chains are constructed using the *Metropolis-Hastings (MH)* algorithm, which at the current state x_t draws a point from a *proposal distribution* $X^* \sim Q(\cdot|X_t)$, and sets $X_{t+1} \leftarrow X^*$ with probability

$$\min\left(1, \frac{\pi(X^*)Q(X_t|X^*)}{\pi(X_t)Q(X^*|X_t)}\right), \tag{2.14}$$

and $X_{t+1} \leftarrow X_t$ otherwise.

Exact-approximate inference

Replacing $\pi(x)$ in (2.14) with an estimator $\hat{\pi}(x)$ that is unbiased, i.e. $\mathbb{E}\,\hat{\pi}(x) = \pi(x)$ where the expectation is over the estimation randomness, results in *pseudo-marginal* MCMC. It is straight-forward to show that the resulting Markov chain retains the same, asymptotically exact, limiting distribution [5, 16].

Random walk (adaptive) Metropolis

One of the simplest (continuous) proposal distributions is a Gaussian, which results in the *random walk Metropolis* algorithm. At position X_t, the proposal is sampled from

$$Q_t(\cdot|X_t) = \mathcal{N}(\cdot \mid X_t, \nu\Sigma), \tag{2.15}$$

for some user-specified covariance Σ and (positive) *step-size* ν.

Clearly, the choice of Σ impacts sampling efficiency, however, choosing

Σ prior to running the algorithm is challenging, as the structure of the target is yet unknown. One idea is to learn the global covariance structure on the fly. Let $\Sigma_t = \Sigma_t(X_0, X_1, \ldots, X_{t-1})$ denote a covariance matrix estimate obtained from the Markov chain history $\{X_i\}_{i=0}^{t-1}$. The *adaptive Metropolis* algorithm [6, 60, 61] replaces Σ with Σ_t in (2.15),

$$Q_t(\cdot|X_t) = \mathcal{N}(\cdot \mid X_t, \nu_t \Sigma_t). \tag{2.16}$$

Adaptive Metropolis therefore considers global covariance structure of the target when proposing moves, which often leads to improved convergence speed.

While adaptive Metropolis was shown to converge to the same correct asymptotic distribution as random walk metropolis under certain conditions, this is not necessarily true in general [6, Section 2]. Therefore, care has to been taken when adaptive proposal mechanisms are used.

Step-size adaptation

One choice for the scaling factor ν in the Gaussian proposal (2.15) is to use a fixed factor $\nu = 2.38/\sqrt{d}$, which was shown to be optimal on Gaussian targets in an asymptotic sense [48]. This result does not generally hold for adaptive Metropolis in (2.16) or for non-Gaussian targets, but it can nevertheless be used as a heuristic. Alternatively, the scale ν can also be replaced with ν_t, which can be adapted at each step, as in [6, Algorithm 4], to obtain an 'asymptotically optimal' the acceptance rate $\alpha^* \approx 0.234$ from Gelman et al. [48] and Rosenthal [97]. A simple rule to tune the step-size such that the current acceptance rate α_t reaches a desired value α^* is the stochastic approximation recursion by Robbins and Monro [92], heavily used by the work of Andrieu and Thoms [6],

$$\log \nu_{t+1} = \log \nu_t + \lambda_{t+1}(\alpha_t - \alpha^*), \tag{2.17}$$

where λ_t is a 'learning-rate' parameter.

Part I

Adaptive Monte Carlo proposals

Chapter 3

Kernel adaptive Metropolis-Hastings

This chapter is based on collaborative work, D. Sejdinovic, H. Strathmann, M. Lomeli, C. Andrieu, and A. Gretton. "Kernel Adaptive Metropolis-Hastings". In: *International Conference for Machine Learning*. 2012.

A kernel adaptive Metropolis-Hastings algorithm is introduced, for the purpose of sampling from a target distribution with strongly non-linear support. The algorithm embeds the trajectory of the Markov chain into a reproducing kernel Hilbert space, such that the feature space covariance of the samples informs the choice of proposal. The procedure is computationally efficient and straightforward to implement, since the RKHS moves can be integrated out analytically: our proposal distribution in the original space is a normal distribution whose mean and covariance depend on where the current sample lies in the support of the target distribution, and adapts to its local covariance structure. Furthermore, the procedure requires neither gradients nor any other higher order information about the target, making it particularly attractive for contexts such as pseudo-marginal MCMC. Kernel Adaptive Metropolis-Hastings outperforms competing fixed and adaptive samplers on multivariate, highly non-linear target distributions, arising in both real-world and synthetic examples.

Chapter outline

Based on the notion of covariance operators in RKHS, c.f. Section 2.1, we describe a sampling strategy for Gaussian measures in the RKHS in Section 3.1, and introduce a cost function for constructing proposal distributions. In Section 3.2, we outline our main algorithm, termed kernel adaptive Metropolis-Hastings (MCMC Kameleon). We provide experimental comparisons with other fixed and adaptive samplers in Section 3.3, where we show superior performance in the context of pseudo-marginal MCMC for Bayesian classification, and on synthetic target distributions with highly non-linear shape.

3.1 Sampling in RKHS

Our approach is based on the idea that the non-linear support of a target density may be learned using kernel principal component analysis (kernel PCA) [8, 110], which is standard linear PCA [21, Chapter 12] on the empirical version of the covariance operator (2.4) in an RKHS,

$$C_X = \frac{1}{n}\sum_{i=1}^{n} k(\cdot, X_i) \otimes k(\cdot, X_i) - \mu_X \otimes \mu_X,$$

computed on a sample $\{X_i\}_{i=1}^{n}$. The empirical covariance operator behaves as expected from its finite dimensional covariance matrix counterpart: applying the tensor product in definition (2.5) gives

$$\langle f, C_X g\rangle_{\mathcal{H}} = \frac{1}{n}\sum_{i=1}^{n} f(X_i)g(X_i) - \left(\frac{1}{n}\sum_{i=1}^{n} f(X_i)\right)\left(\frac{1}{n}\sum_{i=1}^{n} g(X_i)\right),$$

By analogy with algorithms which use linear PCA directions to inform MH proposals [6, Algorithm 8], non-linear PCA directions could be encoded in the proposal construction, as we described in Sejdinovic et al. [104, Appendix]. Alternatively, the approach we pursue here is to focus on a Gaussian measure in the RKHS determined by the empirical covariance operator C_X. This generalises the proposal (2.16) by Haario et al. [60], which considers the Gaussian measure induced by the empirical covariance matrix on the original space.

We next describe the proposal distribution at iteration t of the MCMC chain. We assume that a subset of the chain history, denoted $X = \{X_i\}_{i=1}^n$, $n \le t-1$, is available. Our proposal is constructed by first directly sampling from the density induced by the empirical covariance operator in the RKHS, and then performing a gradient descent in order to find a corresponding location in input space. The gradient step's cost function solely depends on the embedded Markov chain history into the RKHS. We finally add exploration noise.

As we will see later, it is possible to simplify the procedure by integrating out the sampling in the RKHS, the gradient step, and the exploration noise altogether, leading to a closed-form proposal.

3.1.1 Gaussian measure of the covariance operator

We work with the Gaussian measure on the RKHS $\mathcal{H}$ with mean $k(\cdot, X_t)$ and covariance $\nu^2 C_X$, where $X = \{X_i\}_{i=1}^n$ is the subset of the chain history[1]. While there is no analogue of a Lebesgue measure in an infinite dimensional RKHS, it is instructive (albeit with some abuse of notation) to denote this measure in the 'density form'

$$\mathcal{N}(f \mid k(\cdot, X_t), \nu^2 C_X) \propto \exp\left(-\frac{1}{2\nu^2}\left\langle f - k(\cdot, X_t), C_X^{-1}(f - k(\cdot, X_t))\right\rangle_{\mathcal{H}}\right). \tag{3.1}$$

As C_X is a finite-rank operator, this measure is supported only on a finite-dimensional affine space $k(\cdot, X_t) + \mathcal{H}_X$, where $\mathcal{H}_X = \operatorname{span}\{k(\cdot, X_i)\}_{i=1}^n$ is the subspace spanned by the canonical features of X. Conveniently, samples

[1] We assume w.l.o.g. that X contains the first n of all t data, i.e. $\{X_i\}_{i=1}^t \setminus X = \{X_i\}_{n+1}^t$.

from this measure take the form

$$f = k(\cdot, X_t) + \sum_{i=1}^{n} \beta_i \left[k(\cdot, X_i) - \mu_X\right],$$

where

$$\beta \sim \mathcal{N}(0, \frac{\nu^2}{n} I) \tag{3.2}$$

is isotropic. The sample mean is

$$\mathbb{E}\left[f\right] = \mathbb{E}\left[k(\cdot, X_t) + \sum_{i=1}^{n} \beta_i \left[k(\cdot, X_i) - \mu_X\right]\right] = k(\cdot, X_t),$$

and the sample covariance is

$$\begin{aligned}
&\mathbb{E}\left[(f - k(\cdot, X_t)) \otimes (f - k(\cdot, X_t))\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} \beta_i \beta_j \left(k(\cdot, X_i) - \mu_X\right) \otimes \left(k(\cdot, X_j) - \mu_X\right)\right] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}\left[\beta_i \beta_j\right] \left(k(\cdot, X_i) - \mu_X\right) \otimes \left(k(\cdot, X_j) - \mu_X\right) \\
&= \frac{\nu^2}{n} \sum_{i=1}^{n} \left(k(\cdot, X_i) - \mu_X\right) \otimes \left(k(\cdot, X_i) - \mu_X\right) \\
&= \nu^2 C_X,
\end{aligned}$$

which exactly coincides with (3.1).

3.1.2 Obtaining proposals through gradient descent

The RKHS sample $f = k(\cdot, X_t) + \sum_{i=1}^{n} \beta_i \left[k(\cdot, X_i) - \mu_X\right]$ represents the non-linear covariance structure of the Markov chain history, and therefore would be a useful proposal. Unfortunately, this sample does not in general have a corresponding pre-image in the original domain $\mathcal{X} = \mathbb{R}^d$, i.e. there is no point $X_* \in \mathcal{X}$ such that $f = k(\cdot, X_*)$. If there were such a point, then we could use it as a proposal in the original domain, as illustrated

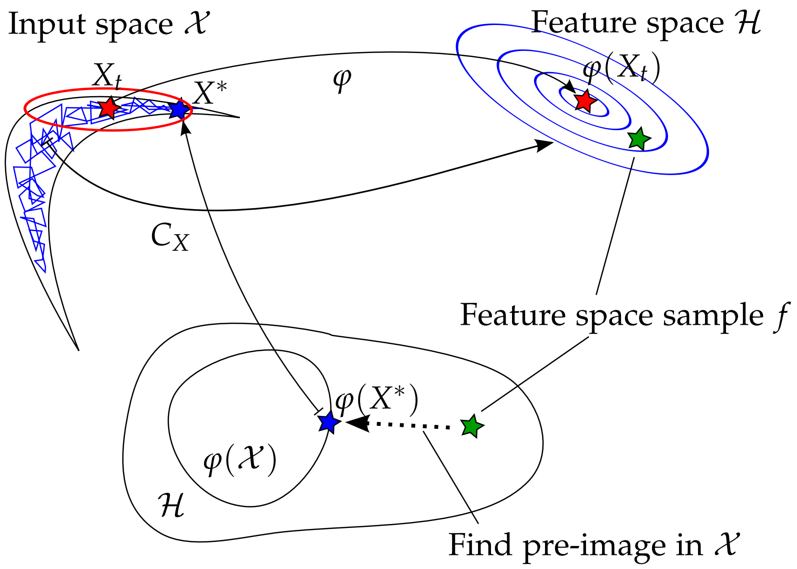


Figure 3.1: Illustration of embedding the Markov chain history into RKHS, sampling from the corresponding empirical Gaussian measure, and taking a gradient step to find the sample's pre-image.

in Figure 3.1. Therefore, we are ideally looking for a point $X^* \in \mathcal{X}$ whose canonical feature map $k(\cdot, X^*)$ is close to f. A natural way to quantify 'closeness' here is the RKHS norm, which allows to translate the problem of finding the pre-image into an optimisation problem,

$$\begin{aligned} &\operatorname*{argmin}_{x\in\mathcal{X}} \|k(\cdot,x)-f\|^2_{\mathcal{H}} \\ =&\operatorname*{argmin}_{x\in\mathcal{X}} \left\{k(x,x)-2k(x,X_t)-2\sum_{i=1}^{n}\beta_i\left[k(x,X_i)-\mu_X(x)\right]\right\} \\ =:&\operatorname*{argmin}_{x\in\mathcal{X}} \{g(x)\}, \end{aligned} \tag{3.3}$$

where we implicitly defined the kernel induced cost function $g(x)$. In general, this is a non-convex minimisation problem, and may be difficult to solve [11]. Rather than solving it for every new vector of coefficients β, which would lead to an excessive computational burden for every proposal made, we instead take a single descent step along the gradient of g

in (3.3). The proposed new point is

$$X^* = X_t - \eta \nabla_x g(x)|_{x=X_t} + \xi,$$

where η is a gradient step-size parameter and $\xi \sim \mathcal{N}(0, \gamma^2 I)$ is an additional isotropic 'exploration' term added *after* the gradient step. This exploration noise avoids the proposal to collapse in unexplored regions, and thereby ensures the proposal to fall back to an isotropic random walk[2].

It is useful to split the scaled gradient at X_t into two terms as

$$\eta \nabla_x g(x)|_{x=X_t} = \eta \left(a_{X_t} - M_{X,X_t} H \beta \right),$$

where $a_{X_t} = \nabla_x k(x,x)|_{x=X_t} - 2\nabla_x k(x, X_t)|_{x=X_t}$,

$$M_{X,X_t} = 2 \left[\nabla_x k(x, X_1)|_{x=X_t}, \ldots, \nabla_x k(x, X_n)|_{x=X_t} \right] \tag{3.4}$$

is a $d \times n$ matrix, and $H = I - \frac{1}{n} \mathbf{1}_{n \times n}$ is the $n \times n$ centering matrix. Figure 3.2 shows $g(x)$ and its gradients for several samples of β-coefficients, in the case where the underlying X-samples are from the two-dimensional nonlinear Banana target distribution of Haario et al. [60] that we will use in Section 3.3.2. It can be seen that g may have multiple local minima, and that it varies most along the high-density regions of the Banana distribution.

3.2 MCMC Kameleon algorithm

We now have a recipe to construct a proposal that is able to adapt to the local covariance structure for the current chain state X_t. While we will simplify this proposal by integrating out the moves in the RKHS, it is instructive to think of the proposal generating process as:

1. Sample $\beta \sim \mathcal{N}(0, \nu^2 I)$, n real-valued RKHS coefficients.

[2] We will see in Chapter 4 that this property translates into the fact that our algorithm converges at least as fast as standard random walks.

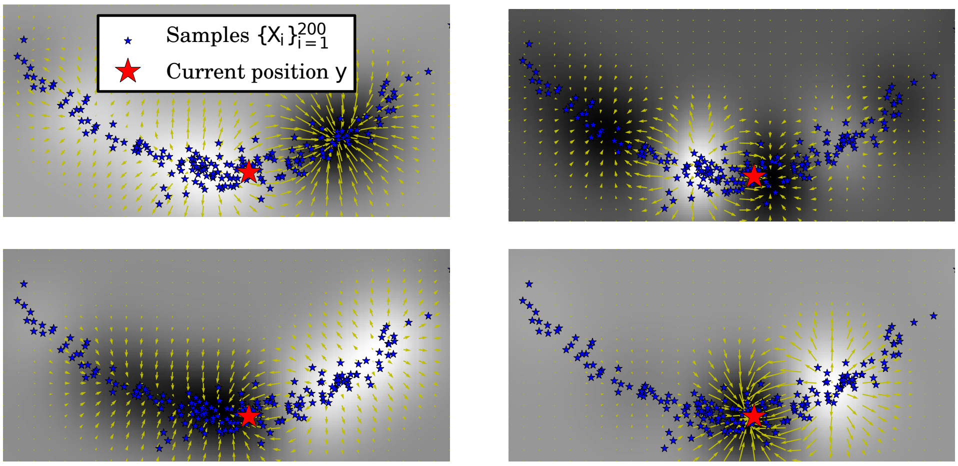


Figure 3.2: Heatmaps (white denotes large) and gradients of $g(x)$ for four samples of β and fixed $\{X_i\}_{i=1}^{n}$ (blue), at the current position $y = X_t$ (red).

This represents an RKHS sample

$$f = k(\cdot, X_t) + \sum_{i=1}^{n} \beta_i \left[k(\cdot, X_i) - \mu_X \right]$$

which induces the cost function $g(x)$.

2. Move along the gradient of g and add noise $\xi \sim \mathcal{N}(0, \gamma^2 I)$, i.e. $X^* = X_t - \eta \nabla_x g(x)|_{x=X_t}$, d-dimensional in the original space

   This gives a proposal

$$X^* | X_t, \beta, \xi \sim \mathcal{N}(X^* \mid X_t - \eta a_{X_t} + \eta M_{X,X_t} H \beta, \gamma^2 I). \tag{3.5}$$

3.2.1 Closed form proposal

As the the explicit sampling of f in the RKHS, the (deterministic) gradient step, and the addition of exploration noise only involve Gaussian distributions, the procedure can be integrated out analytically. This leads to a *closed form* Gaussian proposal whose covariance matrix locally aligns to the target covariance structure observed through the Markov chain history.

The first step in the derivation of the explicit proposal density is to show that as long as k is a differentiable positive definite kernel, the term

a_{X_t} vanishes. Derivations of the following result are given in [104, Appendix].

Proposition 1. *Let k be a differentiable positive definite kernel. Then*

$$a_{X_t} = \nabla_x k(x,x)|_{x=X_t} - 2\nabla_x k(x,X_t)|_{x=X_t} = 0.$$

Since $a_{X_t} = 0$, the gradient step-size η in (3.5) always appears together with β, we merge η and the scale ν of the β-coefficients in (3.2) into a single scale parameter, and set $\eta = 1$ henceforth. Furthermore, since both β and $X^*|X_t,\beta,\xi$ admit multivariate Gaussian densities, the proposal density can be computed via integrating over β and ξ,

$$Q_X(X^*|X_t) = \int p(X^*|X_t,\beta,\xi)p(\beta)p(\xi)\mathrm{d}\beta\mathrm{d}\xi,$$

where $p(X^*|X_t,\beta,\xi)$ is the density of the explicit proposal in (3.5): an isotropic Gaussian whose mean is shifted according to the kernel gradients from (3.4). We arrive at the following closed form expression for the proposal distribution, a simple multivariate Gaussian density.

Proposition 2. $Q_X(X^*|X_t) = \mathcal{N}(X^* \mid X_t, \gamma^2 I + \nu^2 M_{X,X_t} H M_{X,X_t}^\top)$.

Figure 3.3 depicts contours of the proposal distribution $Q_X(\cdot|X_t)$ at various states X_t for a fixed subsample X from various targets (target details in Section 3.3.2).

With the simplified proposal distribution in Proposition 2, we proceed with the standard Metropolis-Hastings accept/reject scheme (2.14), where the proposed sample X^* is accepted with probability

$$\alpha(X_t, X^*) := \min\left\{1, \frac{\pi(X^*)Q_X(X_t|X^*)}{\pi(X_t)Q_X(X^*|X_t)}\right\}, \tag{3.6}$$

giving rise to the MCMC Kameleon algorithm. Each $\pi(X^*)$ and $\pi(X_t)$

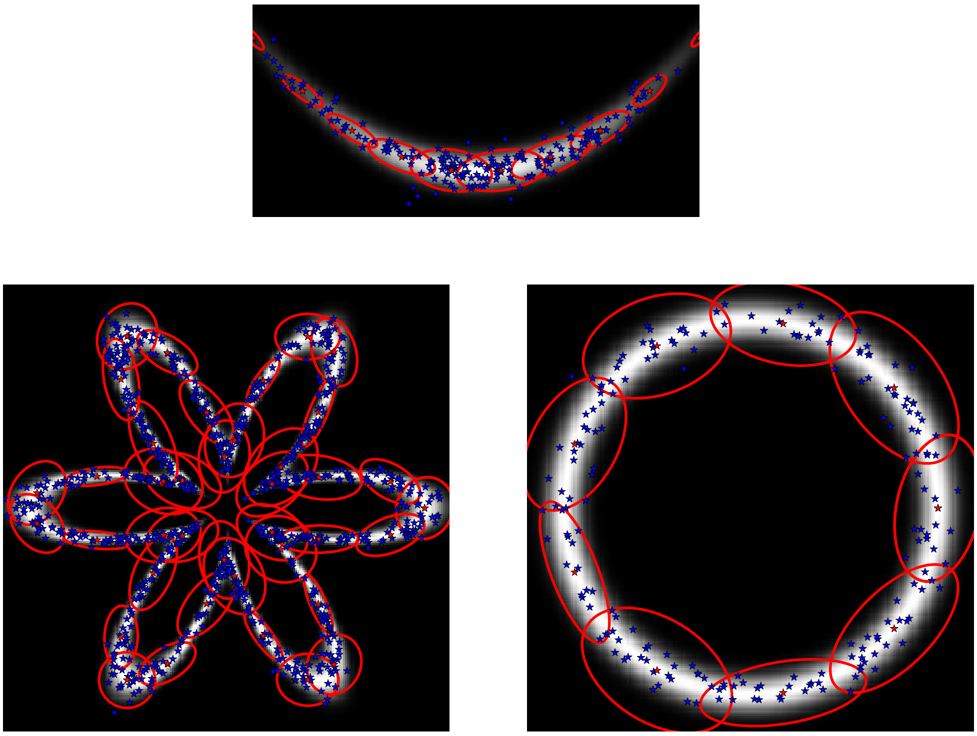


Figure 3.3: 95% contours (red) of proposal distributions evaluated at a number of points, for the first two dimensions of the banana target of Haario et al. [60], and ring and flower. Underneath is the density heatmap, and the samples (blue) used to construct the proposals.

could be replaced by their unbiased estimates without impacting the invariant distribution [5].

MCMC Kameleon
Input: un-normalised target π, subsample size n, scaling parameters ν,γ, adaptation probabilities $\{a_t\}_{t=0}^{\infty}$, kernel k,

- At iteration $t+1$,

    1. With probability a_t, update a random subsample $X=\{X_i\}_{i=1}^{\min(n,t)}$ of the chain history $\{X_i\}_{i=0}^{t-1}$,
    2. Sample proposed point x^* from $Q_X(\cdot|X_t)=\mathcal{N}(\cdot \mid X_t,\gamma^2 I+\nu^2 M_{X,X_t}HM_{X,X_t}^{\top})$, where $M_{X,x_t}$is given in (3.4) and $H=I-\frac{1}{n}\mathbf{1}_{n\times n}$ is the centering matrix,
    3. Accept or reject with the Metropolis-Hastings acceptance probability in (3.6),

$$X_{t+1} \quad \leftarrow \quad \begin{cases} X^*, & \text{with probability } \alpha(X_t,X^*), \\ X_t, & \text{with probability } 1-\alpha(X_t,X^*). \end{cases}$$

3.2.2 Properties of the Algorithm

Update schedule and convergence.

MCMC Kameleon requires a subsample $X=\{X_i\}_{i=1}^{n}$ at each iteration of the algorithm, and the proposal distribution $Q_X(\cdot|X_t)$ is updated each time a new subsample X is obtained. It is well known that a chain which keeps adapting the proposal distribution need not converge to the correct target [6]. To guarantee convergence, we introduce adaptation probabilities $\{a_t\}_{t=0}^{\infty}$, such that $a_t\to 0$ and $\sum_{t=1}^{\infty}a_t=\infty$, and at iteration t we update the subsample X with probability a_t. As adaptations occur with decreasing probability, Roberts and Rosenthal [94] implies that the resulting algorithm is ergodic and converges to the correct target. Another straightforward way to guarantee convergence is to fix the set $X=\{X_i\}_{i=1}^{n}$ after a 'burn-in' phase, i.e. to stop adapting altogether [94, Proposition 2]. In this case, a 'burn-in' phase is used to get a rough sketch of the shape of the distribution: the initial samples need not come from a converged or even valid MCMC chain, and it suffices to have a scheme with good exploratory properties, e.g. Welling and Teh [129]. In MCMC Kameleon, the term γ allows exploration in the initial iterations of the chain (while the subsample X is still not informative about the structure of the target) and provides regularisation of the proposal covariance in cases where it might become ill-conditioned. Intuitively, a good approach to setting γ is to slowly decrease it with each adaptation, such that the learned covariance progressively dominates the proposal. We will return to the question of convergence speed in the case of pre-mature stopping of the update in Chapter 4.

Symmetry

In Haario et al. [61], the proposal distribution is asymptotically symmetric due to the vanishing adaptation property. Therefore, the authors compute the standard Metropolis acceptance probability, i.e. the proposal density ratio in the MH acceptance probability in (3.6) is equal to one. In our case, the proposal distribution is a Gaussian with mean at the current state of the chain X_t and covariance $\gamma^2 I + \nu^2 M_{X,X_t} H M_{X,X_t}^\top$, where M_{X,X_t} depends both on the current state X_t and a random subsample $X = \{X_i\}_{i=1}^n$ of the chain history $\{X_i\}_{i=0}^{t-1}$. This proposal distribution is never symmetric (as covariance of the proposal always depends on the current state of the chain), and therefore we use the Metropolis-Hastings acceptance probability to reflect this.

3.2.3 Examples of covariance structure for standard kernels

The proposal distributions in MCMC Kameleon are dependent on the choice of the kernel k. To gain intuition regarding their covariance structure, we give two examples.

Linear kernel

In the case of a linear kernel $k(x,x') = x^\top x'$, we obtain

$$M_{X,X_t} = 2\left[\nabla_x x^\top X_1|_{x=X_t}, \ldots, \nabla_x x^\top X_n|_{x=X_t}\right] = 2X^\top,$$

so the proposal is given by

$$Q_X(\cdot|X_t) = \mathcal{N}(X_t, \gamma^2 I + 4\nu^2 X^\top H X).$$

Thus, the proposal uses the scaled empirical covariance $X^\top HX$, with an additional isotropic exploration component, and depends on X_t only through the mean. MCMC Kameleon therefore is a generalisation of the adaptive Metropolis algorithm [60].

Gaussian kernel

In the case of a Gaussian kernel $k(x,x') = \exp\left(-\frac{\|x-x'\|_2^2}{2\sigma^2}\right)$, since $\nabla_x k(x,x') = \frac{1}{\sigma^2}k(x,x')(x'-x)$, we obtain

$$M_{X,X_t} = \frac{2}{\sigma^2}\left[k(X_t,X_1)(X_1 - X_t), \ldots, k(X_t,X_n)(X_n - X_t)\right].$$

Consider how this encodes the covariance structure of the target distribution, with the (i,j)-th entry given by

$$\begin{aligned}\gamma^2\delta_{i=j} \quad &+ \quad \frac{4\nu^2(n-1)}{\sigma^4 n}\sum_{a=1}^{n}\left[k(X_t,X_a)\right]^2(X_{a,i} - X_{t,i})(X_{a,j} - X_{t,j}) \\ &- \quad \frac{4\nu^2}{\sigma^4 n}\sum_{a\neq b} k(X_t,X_a)k(X_t,X_b)(X_{a,i} - X_{t,i})(X_{b,j} - X_{t,j}).\end{aligned}$$

As the first two terms dominate, the previous points X_a which are close to the current state X_t (for which $k(X_t,X_a)$ is large) have larger weights, and thus they have more influence in determining the covariance of the proposal at X_t.

3.3 Experiments

In the experiments, we compare the following samplers:

- **(SM)** Standard Metropolis with the isotropic proposal $Q(\cdot|X_t) = \mathcal{N}(X_t, \nu^2 I)$ and scaling $\nu = 2.38/\sqrt{d}$
- **(AM-FS)** Adaptive Metropolis with a learned covariance matrix and fixed scaling $\nu = 2.38/\sqrt{d}$
- **(AM-LS)** Adaptive Metropolis with a learned covariance matrix and scaling learned to bring the acceptance rate close to $\alpha^* = 0.234$ using (2.17).
- **(KAMH-LS)** MCMC Kameleon with the scaling ν learned in the same fashion (γ was fixed to 0.2), and which also stops adapting the proposal after the burn-in of the chain

In all experiments, we use a random sub-sample X of the Markov chain history of size $n = 1000$, and a Gaussian kernel with bandwidth selected according to the median heuristic[3]. We consider the following non-linear targets:

- the posterior distribution of Gaussian Process classification hyperparameters [44] on the UCI glass dataset
- the synthetic banana-shaped distribution of Haario et al. [60] and a flower-shaped distribution concentrated on a circle with a periodic perturbation

3.3.1 Pseudo-marginal MCMC for Bayesian classification

In the first experiment, we illustrate usefulness of the MCMC Kameleon sampler in the context of Bayesian classification with GPs [131]. Variants of this experiment will be used in Chapter 4, Chapter 5, and Chapter 6.

[3] The median heuristic matches the bandwidth of a Gaussian kernel with the median pair-wise distance within the training data.

Set-up

Consider the joint distribution of latent function responses $f \in \mathbb{R}^n$, labels $y \in \{-1,1\}^n$, and hyper-parameters θ, given by

$$p(f,y,\theta) = p(\theta)p(f|\theta)p(y|f),$$

where $f|\theta \sim \mathcal{N}(0,\mathcal{K}_\theta)$, with $\mathcal{K}_\theta$ modelling the covariance between latent function values evaluated at d-dimensional input covariates x_a,x_b:

$$(\mathcal{K}_\theta)_{ab} = \kappa(x_a,x_b|\theta) = \exp\left(-\frac{1}{2}\sum_{i=1}^{d}\frac{(x_{a,i}-x_{b,i})^2}{\ell_i^2}\right)$$

and $\theta_i = \log \ell_i^2$. We restrict our attention to the binary logistic classifier, i.e. the likelihood is given by

$$p(y_i|f_i) = \frac{1}{1-\exp(-y_i f_i)},$$

where $y_i \in \{-1,1\}$. We pursue a fully Bayesian treatment, and estimate the posterior of the hyper-parameters θ. As observed by Murray and Adams [84], a Gibbs sampler on $p(\theta,f|y)$, which samples from $p(f|\theta,y)$ and $p(\theta|f,y)$ in turn, is problematic, as $p(\theta|f,y)$ is extremely sharp, drastically limiting the amount that any Markov chain can update $\theta|f,y$. On the other hand, if we directly consider the marginal posterior $p(\theta|y) \propto p(y|\theta)p(\theta)$ of the hyper-parameters, a much less peaked distribution can be obtained. The marginal likelihood $p(y|\theta)$, however, is intractable for the non-Gaussian likelihood $p(y|f)$, so it is not possible to analytically integrate out the latent variables. Pseudo-marginal MCMC methods, c.f. Andrieu and Roberts [5] and Section 2.3, enable *asymptotically exact* inference on this problem, by replacing $p(y|\theta)$ with an unbiased estimate. Such an estimate

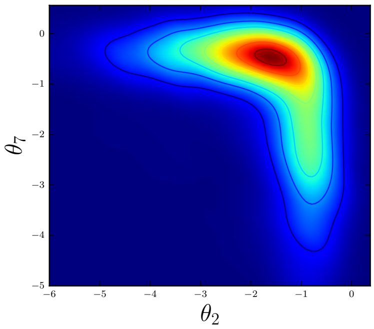


Figure 3.4: Dimensions 2 and 7 of the marginal hyperparameter posterior on the UCI Glass dataset

can for example be obtained via importance sampling,

$$\hat{p}(y|\theta) := \frac{1}{n_{\text{imp}}} \sum_{i=1}^{n_{\text{imp}}} p(y|f^{(i)}) \frac{p(f^{(i)}|\theta)}{q(f^{(i)}|\theta)},$$

where $\left\{ f^{(i)} \right\}_{i=1}^{n_{\text{imp}}} \sim q(f|\theta)$ are n_{imp} importance samples form a tractable distribution $q(f|\theta)$

Filippone and Girolami [44] applied this idea to the presented GP classification framework and obtained state-of-the-art results in terms of accuracy of predictions and uncertainty quantification. The importance distribution $q(f|\theta)$ is chosen as the Laplace or as the expectation propagation (EP) approximation of $p(f|y,\theta) \propto p(y|f)p(f|\theta)$.

Data

We consider the UCI glass dataset [9], where classification of window against non-window glass is sought. Due to the heterogeneous structure of each of the classes (i.e. non-window glass consists of containers, tableware and headlamps), there is no single consistent set of length-scales determining the decision boundary, so one expects the posterior of the covariance bandwidths θ_d to have a complicated (non-linear) shape. This is illustrated by the plot of the posterior projections to the dimensions 2 and 7 (out of 9) in Figure 3.4.

Experimental protocol

Since the ground truth for the hyper-parameter posterior is not available, we initially run 30 standard Metropolis chains for 500,000 iterations (with a 100,000 burn-in), keep every 1000-th sample in each of the chains, and combine them. The resulting samples are used as a benchmark, to evaluate the performance of shorter single-chain runs of **SM**, **AM-FS**, **AM-LS** and **KAMH-LS**. We execute each of these algorithms for 100,000 iterations (with a 20,000 burn-in) and keep every 20-th sample.

Results

We use two metrics for evaluating the performance, always relative to the large-scale benchmark sample. First, we compute the Euclidean distance of empirical mean of the sampler output $\hat{\mu}_\theta$ and the mean of the benchmark samples μ_θ^b,

$$\left\|\hat{\mu}_\theta - \mu_\theta^b\right\|_2,$$

as a function of sample size (Figure 3.5, left). Second, in order to quantify convergence in higher order moments, we estimate the MMD using a third order polynomial kernel, c.f. Section 2.1, between each sampler output and the benchmark sample (Figure 3.5, right). The figures indicate that **KAMH-LS** approximates the benchmark sample better than the competing approaches, where the effect is especially pronounced in the high order moments, indicating that **KAMH-LS** thoroughly explores the distribution support in a relatively small number of samples.

Computational costs

The bulk of the cost for pseudo-marginal MCMC is in importance sampling in order to obtain the acceptance ratio. Therefore, the additional cost imposed by **KAMH-LS** is negligible. Indeed, we observed that there is an increase of only 2-3% in terms of effective computation time in comparison to all other samplers, for the chosen size of the chain history sub-sample ($n = 1000$).

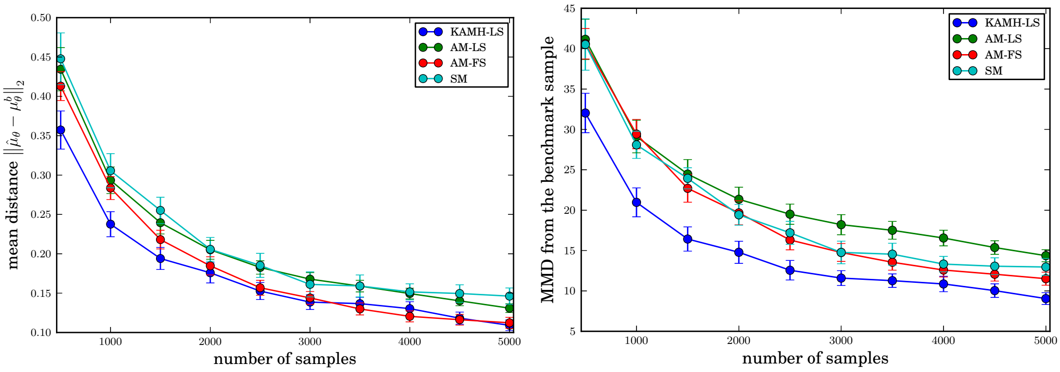


Figure 3.5: The comparison of **SM**, **AM-FS**, **AM-LS** and **KAMH-LS** in terms of the distance between the estimated mean and the mean on the benchmark sample (left) and in terms of the maximum mean discrepancy to the benchmark sample (right). The results are averaged over 30 chains for each sampler. Error bars represent 80%-confidence intervals.

3.3.2 Synthetic examples.

We next evaluate MCMC Kameleon on a number of artificial targets with high degrees of non-linearity. In these examples, exact quantile regions of the targets can be computed analytically, so we can directly assess performance without the need to estimate distribution distances on the basis of samples (i.e. by estimating MMD to the benchmark sample). We compute the following measures of performance (similarly as in Andrieu and Thoms [6] and Haario et al. [60]) based on the chain after burn-in: average acceptance rate, norm of the empirical mean (the true mean is by construction zero for all targets), and the deviation of the empirical quantiles from the true quantiles.

Banana and flower target

In Haario et al. [60], the following family of non-linear target distributions is considered. Let $X \sim \mathcal{N}(0,\Sigma)$ be a multivariate normal in $d \geq 2$ dimensions, with $\Sigma = \text{diag}(v,1,\ldots,1)$, which undergoes the transformation $X \to Y$, where $Y_2 = X_2 + b(X_1^2 - v)$, and $Y_i = X_i$ for $i \neq 2$. We will write

$Y \sim \mathcal{B}(b,v)$. It is clear that $\mathbb{E}Y = 0$, and that

$$\mathcal{B}(y;b,v) = \mathcal{N}(y_1;0,v)\mathcal{N}(y_2;b(y_1^2 - v),1)\prod_{j=3}^{d}\mathcal{N}(y_j;0,1).$$

The second target distribution we consider is the d-dimensional flower target $\mathcal{F}(r_0,A,\omega,\sigma)$, with

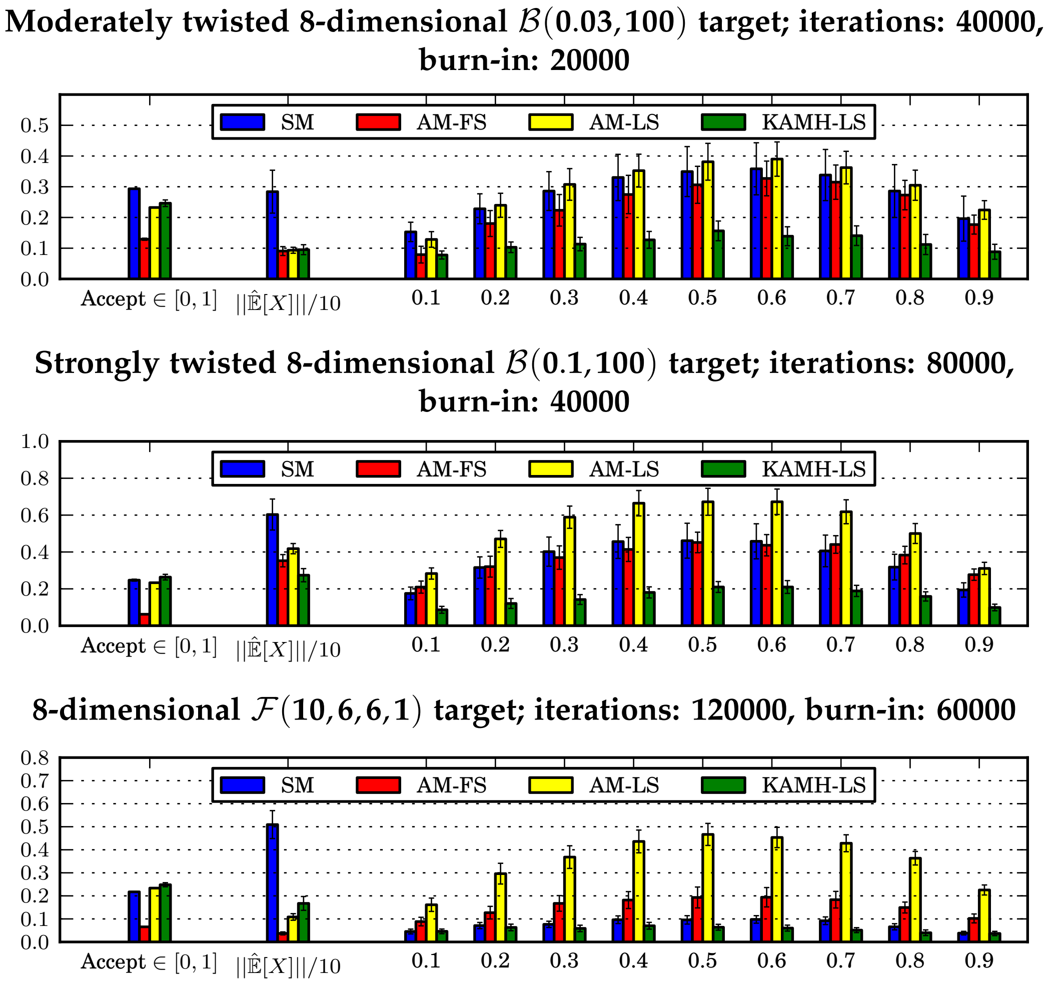
$$\begin{aligned}\mathcal{F}(x;r_0,A,\omega,\sigma) = \\ \exp\left(-\frac{\sqrt{x_1^2+x_2^2} - r_0 - A\cos\left(\omega \text{atan2}\left(x_2,x_1\right)\right)}{2\sigma^2}\right) \\ \times\prod_{j=3}^{d}\mathcal{N}(x_j;0,1).\end{aligned}$$

This distribution concentrates around the r_0-circle with a periodic perturbation (with amplitude A and frequency ω) in the first two dimensions.

Results

We consider 8-dimensional target distributions: the moderately twisted $\mathcal{B}(0.03,100)$ banana target (Figure 3.6, top) and the strongly twisted $\mathcal{B}(0.1,100)$ banana target (Figure 3.6, middle) and $\mathcal{F}(10,6,6,1)$ flower target (Figure 3.6, bottom).

The results show that MCMC Kameleon is superior to the competing samplers. Since the covariance of the proposal adapts to the local structure of the target at the current chain state, as illustrated in Figure 3.3, MCMC Kameleon does not suffer from wrongly scaled proposal distributions. The result is a significantly improved quantile performance in comparison to all competing samplers, as well as a comparable or superior accuracy in estimating the norm of the empirical mean. **SM** has a significantly larger norm of the empirical mean, due to its purely random walk behaviour (e.g. the chain tends to get stuck in one part of the space, and is not able to traverse both tails of the banana target equally well). **AM** with fixed scale has a low acceptance rate (indicating that the scaling of the proposal is too large), and even though the norm of the empirical mean is much closer to the true value, quantile performance of the chain is poor. Even if the estimated covariance matrix closely resembles the true global covariance matrix of the target, using it to construct proposal distributions at every state of the chain may not be the best choice. For example, **AM** correctly captures the scaling along individual dimensions for the flower target (the norm of its empirical mean is close to its true value of zero) but fails to capture local dependence structure. The flower target, due to its symmetry, has an isotropic covariance in the first two dimensions – even though they are highly dependent. This leads to a mismatch in the scale of the covariance and the scale of the target, which concentrates on a thin band in the joint space. **AM-LS** has the 'correct' acceptance rate, but the quantile performance is even worse, as the scaling now becomes too small to traverse high-density regions of the target. Figure 3.7 illustrates how the norm of the mean and quantile deviation (shown for 0.5-quantile) decrease as a function of the number of iterations. This shows that the observed trends persist along the evolution of the whole chain.



Moderately twisted 8-dimensional $\mathcal{B}(0.03,100)$ target; iterations: 40000, burn-in: 20000

Strongly twisted 8-dimensional $\mathcal{B}(0.1,100)$ target; iterations: 80000, burn-in: 40000

8-dimensional $\mathcal{F}(10,6,6,1)$ target; iterations: 120000, burn-in: 60000

Figure 3.6: Results for three non-linear targets, averaged over 20 chains for each sampler. *Accept* is the acceptance rate scaled to the interval [0,1]. The norm of the mean $||\hat{\mathbb{E}}[X]||$ is scaled by 1/10 to fit into the figure scaling, and the bars over the 0.1,...,0.9-quantiles represent the deviation from the exact quantiles, scaled by 10, i.e. 0.1 corresponds to 1% deviation. Error bars represent 80%-confidence intervals.

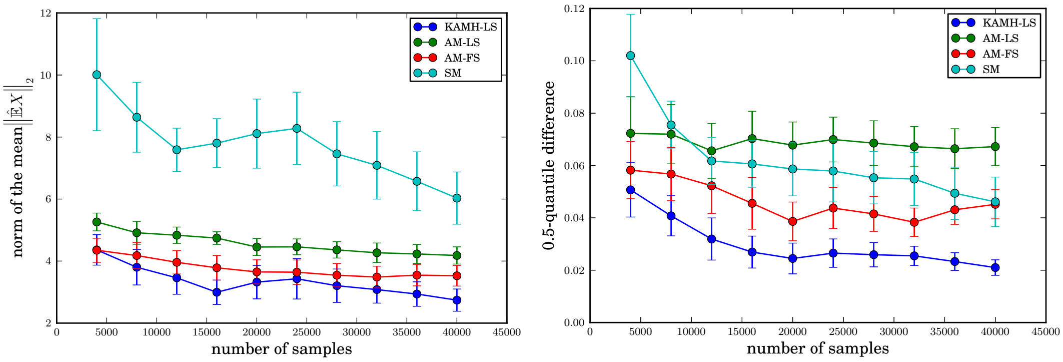


Figure 3.7: Comparison of **SM**, **AM-FS**, **AM-LS** and **KAMH-LS** in terms of the norm of the estimated mean (left) and in terms of the deviation from the 0.5-quantile (right) on the strongly twisted Banana distribution. The results are averaged over 20 chains for each sampler. Error bars represent 80%-confidence intervals.

Chapter 4

Gradient-free Hamiltonian Monte Carlo with efficient kernel exponential families

This chapter is based on collaborative work, H. Strathmann, D. Sejdinovic, S. Livingstone, Z. Szabo, and A. Gretton. "Gradient-free Hamiltonian Monte Carlo with Efficient Kernel Exponential Families". In: *Advances in Neural Information Processing Systems*. 2015.

We propose *Kernel Hamiltonian Monte Carlo (KMC)*, a gradient-free adaptive MCMC algorithm based on Hamiltonian Monte Carlo (HMC). On target densities where classical HMC is not an option due to intractable gradients, KMC adaptively learns the target's gradient structure by fitting an exponential family model in a reproducing kernel Hilbert space. Computational costs are reduced by two novel efficient approximations to this gradient. While being asymptotically exact, KMC mimics HMC in terms of sampling efficiency, and offers substantial mixing improvements over state-of-the-art gradient-free samplers. We support our claims with experimental studies on both toy and real-world applications, including approximate Bayesian computation and exact-approximate MCMC.

Relationship to kernel adaptive Metropolis-Hastings

In Chapter 3, we have seen how adaptively modelling mean and covariance in an RKHS can be used to adapt MCMC proposals to local covariance structure of the unknown underlying target density. It would be desirable to both incorporate prior parametric knowledge about the target density, such as tail behaviour; and to exploit higher order information of the kernel model, such as gradients, in order to propose more efficient moves. Furthermore, the 'Gaussian in feature space' model (3.1) from Chapter 3 currently lacks a theoretical framework regarding consistency, convergence, and generalisation error. This is in particular true for the model's score function.

In this chapter, we extend the ideas from Chapter 3 and directly estimate the target gradients in order to construct a Metropolis-Hastings proposal that mimics HMC and inherits the desirable property of covering wide parts of the target's support at high acceptance probabilities (c.f. Section 1.3.1).

Score estimation

Due to the many unknowns in the model (3.1), we instead use the infinite dimensional exponential family model (2.6), c.f. Section 2.1. As we will see, conveniently, the MH accept/reject step (2.14) does not require access to the (unavailable) normalising constant of the model used for the proposal construction. Furthermore, the estimator enjoys theoretical guarantees [117], and allows to include prior parametric information via the base measure q_0 in (2.6). More importantly, the method has been empirically observed to be relatively robust to increasing dimensionality – in sharp contrast to classical kernel density estimation [127, Section 6.5].

The need for approximations

As mentioned in Section 2.1, if we used the original estimator (2.11), (2.13), computational costs would grow cubically in dimension and number of samples, and fitting the model would quickly become infeasible.

We therefore develop two novel approximations to the infinite dimensional exponential family model. The first approximation, *score matching lite*, is based on computing the solution in terms of a lower dimensional, yet growing, subspace in the RKHS. As we will see in Section 4.3, KMC with score matching lite (*KMC lite*) is geometrically ergodic on the same class of targets as standard random walks. The second approximation uses a finite dimensional feature space (*KMC finite*) with a random Fourier basis [89]. KMC finite is an efficient online estimator that allows to use *all* of the Markov chain history, at the cost of decreased efficiency in unexplored regions. A choice between KMC lite and KMC finite ultimately depends on the ability to initialise the sampler within high-density regions of the target; alternatively, the two approaches could be combined.

Problem set-up

In this chapter we assume that the domain of interest $\mathcal{X}$ is a compact[1] subset of $\mathbb{R}^d$. Again, denote the un-normalised *target* density on $\mathcal{X}$ by π. Recall we are interested in constructing a Markov chain $X_1 \to X_2 \to \dots$ such that $\lim_{t\to\infty} X_t \sim \pi$, c.f. Section 2.3.

Following Chapter 3, we assume that π is intractable, i.e. we can neither evaluate $\pi(\cdot)$ nor[2] $\nabla \log \pi(\cdot)$, but can only estimate it without bias via $\hat{\pi}(\cdot)$. Again, replacing $\pi(\cdot)$ with $\hat{\pi}(\cdot)$ results in pseudo-marginal MCMC [5, 16], which asymptotically remains exact. This leaves random-walk based methods as the state-of-the-art, c.f. Chapter 3 and [6]. We aim to overcome random-walk behaviour with the use of Hamiltonian dynamics, so as to obtain significantly more efficient sampling [86].

[1] The compactness restriction is imposed to satisfy the assumptions in [117], i.e. for the density model estimator.

[2] Throughout the chapter ∇ denotes the gradient operator w.r.t. x.

Hamiltonian Monte Carlo

HMC uses deterministic, measure-preserving maps to propose distant, uncorrelated moves with a high acceptance probability [86]. Starting from the current Markov chain state $q = X_t$ whose density is the negative log target, referred to as the *potential energy*

$$U(q) := -\log \pi(q),$$

we introduce an auxiliary *momentum* variable $p \sim \exp(-K(\cdot))$ with $p \in \mathcal{X}$; usually with a 'Gaussian' quadratic form

$$K(p) = \frac{1}{2} p^\top M p$$

for some positive definite M. The joint distribution of (p,q) is then proportional to $\exp\left(-H(p,q)\right)$, where

$$H(p,q) := K(p) + U(q)$$

is called the *Hamiltonian*. $H(p,q)$ defines a *Hamiltonian flow*, parametrised by a trajectory length $t \in \mathbb{R}$, which is a map $\phi_t^H : (p,q) \mapsto (p^*,q^*)$ for which $H(p^*,q^*) = H(p,q)$. This allows constructing π-invariant Markov chains: for a chain at state $q = X_t$, repeatedly

(i) re-sample $p' \sim \exp(-K(\cdot))$ (independently of previous p)

(ii) apply the Hamiltonian flow for time t, giving $(p^*,q^*) = \phi_t^H(p',q)$.

The flow can be generated by

$$\frac{\mathrm{d}p}{\mathrm{d}t} = -\frac{\partial H}{\partial q} = -\frac{U}{\partial q} \qquad \frac{\mathrm{d}q}{\mathrm{d}t} = \frac{\partial H}{\partial p} = \frac{K}{\partial p}. \tag{4.1}$$

In practice, (4.1) is usually unavailable and we need to resort to approximations. In this work, we limit ourselves to the leap-frog integrator; see [86] for details. To correct for discretisation error, a Metropolis acceptance procedure can be applied: starting from (p',q), the end-point of the approximate trajectory is accepted with probability

$$\min\left[1,\exp\left(-H(p^*,q^*)+H(p',q)\right)\right].$$

It is clear that computing $(p^*,q^*)=\phi_t^H(p',q)$ via (4.1) requires access to $\nabla\log\pi$, a fact that causes classical HMC to be unavailable in this context.

4.1 Kernel Hamiltonian dynamics

We now replace the potential energy in (4.1) by a kernel induced surrogate computed from the history of the Markov chain. This surrogate does not require gradients of the log-target density. The surrogate induces a kernel Hamiltonian flow, which can be numerically simulated using standard leap-frog integration. As with the discretisation error in HMC, any deviation of the kernel induced flow from the true flow is corrected via a Metropolis acceptance procedure. As in the pseudo-marginal MCMC approach the acceptance ratio here contains the estimation noise from $\hat{\pi}$ and in particular re-uses previous values of $\hat{\pi}$, c.f. [5, Table 1]. Consequently, the stationary distribution of the chain remains correct, given that we take care when adapting the surrogate.

We construct a kernel induced potential energy surrogate whose gradients approximate the gradients of the true potential energy U in (4.1), without accessing π or $\nabla\pi$ directly, but only using the history of the Markov chain. To that end, we model the (un-normalised) target density $\pi(x)$ with the infinite dimensional exponential family model (2.6), i.e.

$$\nabla f\approx-\nabla U=\nabla\log\pi$$

where f is the RKHS function corresponding to the model.

We return to estimation in Section 4.2 where we develop two efficient approximations, and again in Chapter 6 where we develop an approximation with guarantees. For now, assume access to an $\hat{f} \in \mathcal{H}$ such that $\nabla \hat{f}(x) \approx \nabla \log \pi(x)$.

Kernel induced Hamiltonian flow

We define a kernel induced Hamiltonian operator by replacing U in the potential energy part $\frac{\partial U}{\partial p}\frac{\partial}{\partial q}$ in (4.1) by our kernel surrogate $U_k = f$. It is clear that depending on U_k, the resulting kernel induced Hamiltonian $H_k(p,q) = K(p) + U_k(q)$ differs from the original one. That said, any bias on the resulting Markov chain, in addition to discretisation error from the leapfrog integrator, is naturally corrected for in the Metropolis step. We accept an end-point $\phi_t^{H_k}(p',q)$ of a trajectory starting at (p',q) along the *kernel induced* flow with the HMC version of the MH acceptance probability (2.14)

$$\min\left[1, \exp\left(-H\left(\phi_t^{H_k}(p',q)\right) + H(p',q)\right)\right], \tag{4.2}$$

where $H\left(\phi_t^{H_k}(p',q)\right)$ corresponds to the *true* Hamiltonian evaluated at the *kernel induced* proposal $\phi_t^{H_k}(p',q)$. Here, in the pseudo-marginal context, we replace both terms in the ratio in (4.2) by unbiased estimates, i.e. we replace $\pi(q)$ within H with an unbiased estimator $\hat{\pi}(q)$. This also involves 'recycling' the estimates of H from previous iterations to ensure asymptotic correctness, c.f. [5, Table 1]. Any deviations of the kernel induced flow from the true flow result in a decreased acceptance probability (4.2). We therefore need to control the approximation quality of the kernel induced potential energy to maintain high acceptance probability in practice. See Figure 4.1 for an illustrative example.

4.2 Two estimators for kernel exponential families

We now return to estimating the infinite dimensional exponential family model from data. The original estimator in [117] has a large computational cost. This is problematic in the adaptive MCMC context, where the

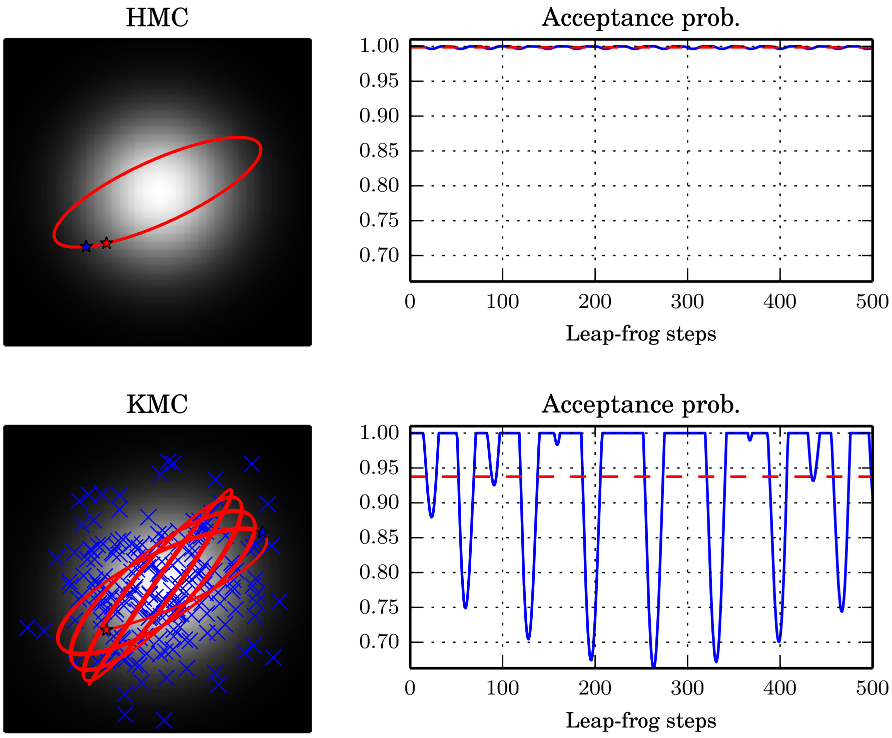


Figure 4.1: Hamiltonian trajectories on a 2-dimensional standard Gaussian. End points of such trajectories (red stars to blue stars) form the proposal of HMC-like algorithms. **Top:** Plain Hamiltonian trajectories oscillate on a stable orbit, and acceptance probability is close to one. **Bottom:** Kernel induced trajectories and acceptance probabilities on an estimated energy function.

model has to be updated on a regular basis. We propose two efficient approximations, each with its strengths and weaknesses. Both are based on score matching, c.f. Section 2.2, and the kernel exponential family model, c.f. Section 2.1.

4.2.1 Infinite dimensional exponential families lite

The original estimator of f in (2.11) takes a dual form in an RKHS subspace spanned by nd kernel derivatives, c.f. [117, Theorem 4] and (2.6). The update of the proposal at the iteration t of MCMC requires inversion of a $td \times td$ matrix. This is clearly prohibitive if we are to run even a moderate number of iterations of a Markov chain. Following the ideas in Chapter 3, we take a simple approach to avoid prohibitive computational costs in t: we form a proposal using a random sub-sample of fixed size n from the Markov chain history[3], $X = \{X_i\}_{i=1}^n \subset \{X_i\}_{i=1}^t$. In order to avoid excessive computation when d is large, we replace the full dual solution with a solution in terms of

$$\operatorname{span}\left(\{k(X_i, \cdot)\}_{i=1}^n\right),$$

which covers the support of the true density by construction, and grows with increasing n. That is, we assume that the RKHS function of the model (2.6) takes the 'light' form

$$f(x) = \sum_{i=1}^n \alpha_i k(X_i, x), \tag{4.3}$$

where $\alpha \in \mathbb{R}^n$ are real valued parameters that are obtained by minimising the empirical score matching objective (2.8). This representation is of a form similar to [64, Section 4.1], the main differences being that the basis functions are chosen randomly, the basis set grows with n, and we will require an additional regularising term. Our estimator is summarised in

[3] As in Chapter 3, we assume w.l.o.g. that X contains the first n of all t data.

the following result; a derivation is provided in Section 4.5.

Proposition 3. *Given a set of samples $\{X_i\}_{i=1}^n$ and assuming $f(x) = \sum_{i=1}^n \alpha_i k(X_i, x)$ for the Gaussian kernel $k(x,y) = \exp\left(-\sigma^{-1}\|x-y\|_2^2\right)$, and $\lambda > 0$, the unique minimiser of the $\frac{1}{2}n\sigma^2\lambda\|\alpha\|_2^2$-regularised empirical score matching objective (2.8) is given by*

$$\hat{\alpha}_\lambda = -\frac{\sigma}{2}(C+\lambda I)^{-1}b, \tag{4.4}$$

where $b \in \mathbb{R}^n$ and $C \in \mathbb{R}^{n\times n}$ are given by

$$b = \sum_{\ell=1}^d \left(\frac{2}{\sigma}(Ks_\ell + D_{s_\ell}K\mathbf{1} - 2D_{x_\ell}Kx_\ell) - K\mathbf{1}\right)$$
$$C = \sum_{\ell=1}^d \left[D_{x_\ell}K - KD_{x_\ell}\right]\left[KD_{x_\ell} - D_{x_\ell}K\right],$$

with entry-wise products $s_\ell := (x_\ell \odot x_\ell) \in \mathbb{R}^n$ where $x_\ell = ((X_1)_\ell, \ldots, (X_n)_\ell) \in \mathbb{R}^n$ and $D_x := diag(x) \in \mathbb{R}^{n\times n}$.

The estimator costs $\mathcal{O}(n^3 + dn^2)$ computation (for computing C, b, and for inverting C) and $\mathcal{O}(n^2)$ storage, for a fixed random chain history subsample size n. We describe how this could be further reduced via low-rank approximations to the kernel matrix and conjugate gradient methods in Strathmann et al. [120, Appendix].

Gradients of the model are given as $\nabla f(x) = \sum_{i=1}^n \alpha_i \nabla k(x, X_i)$, i.e. they simply require to evaluate gradients of the kernel function. Evaluation and storage of $\nabla f(\cdot)$ both cost $\mathcal{O}(dn)$.

4.2.2 Exponential families in finite feature spaces

Instead of fitting an infinite-dimensional model on a subset of the available data, the second estimator is based on fitting a finite dimensional approximation using *all* available data $\{X_i\}_{i=1}^t$, in *primal* form. As we will see, updating the estimator when a new data point arrives can be done online.

Define an m-dimensional approximate feature space $\mathcal{H}_m = \mathbb{R}^m$, and denote by $\phi_x \in \mathcal{H}_m$ the embedding of a point $x \in \mathcal{X} = \mathbb{R}^d$ into $\mathcal{H}_m = \mathbb{R}^m$. Assume that the embedding approximates the kernel function as a finite rank expansion $k(x,y) \approx \phi_x^\top \phi_y$. The log un-normalised density of the infinite model (2.6) can be approximated by assuming the model takes the form

$$f(x) = \langle \theta, \phi_x \rangle_{\mathcal{H}_m} = \theta^\top \phi_x \tag{4.5}$$

To fit $\theta \in \mathbb{R}^m$, we again minimise the score matching objective (2.8), as derived in Section 4.6.

Proposition 4. *Given a set of samples $\{X_i\}_{i=1}^t$ and assuming $f(x) = \theta^\top \phi_x$ for a finite dimensional feature embedding $x \mapsto \phi_x \in \mathbb{R}^m$, and $\lambda > 0$, the unique minimiser of the $\lambda \|\theta\|_2^2$-regularised empirical score matching objective (2.8) is given by*

$$\hat{\theta}_\lambda := (C + \lambda I)^{-1} b,$$

where

$$b := -\frac{1}{n} \sum_{i=1}^t \sum_{\ell=1}^d \ddot{\phi}_{X_i}^\ell \in \mathbb{R}^m, \qquad C := \frac{1}{n} \sum_{i=1}^t \sum_{\ell=1}^d \dot{\phi}_{X_i}^\ell \left(\dot{\phi}_{X_i}^\ell \right)^\top \in \mathbb{R}^{m \times m},$$

with $\dot{\phi}_x^\ell := \frac{\partial}{\partial x_\ell} \phi_x$ and $\ddot{\phi}_x^\ell := \frac{\partial^2}{\partial x_\ell^2} \phi_x$.

An example feature embedding based on random Fourier features [89, 116] and a standard Gaussian kernel is

$$\phi_x = \sqrt{\frac{2}{m}} \left[\cos(\omega_1^\top x + u_1), \ldots, \cos(\omega_m^\top x + u_m) \right],$$

with $\omega_i \sim \mathcal{N}(\omega)$ and $u_i \sim \texttt{Uniform}[0, 2\pi]$; more details for this example are given in Section 4.6.2.

The estimator has a one-off cost of $\mathcal{O}(tdm^2 + m^3)$ computation and $\mathcal{O}(m^2)$ storage. Given that we have computed a solution based on the Markov chain history $\{X_i\}_{i=1}^t$, however, it is straightforward to update

C, b, and the solution $\hat{\theta}_\lambda$ online, after a new point X_{t+1} arrives. This is achieved by storing running averages and performing low-rank updates of matrix inversions, and costs $\mathcal{O}(dm^2)$ computation and $\mathcal{O}(m^2)$ storage, *independent* of t. Further details and an algorithmic description are given in Section 4.6.3.

Gradients of the model are $\nabla f(x) = [\nabla \phi_x]^\top \hat{\theta}$, i.e. they require the evaluation of the gradient of the feature space embedding, costing $\mathcal{O}(md)$ computation and and $\mathcal{O}(m)$ storage.

4.3 Kernel Hamiltonian Monte Carlo

By constructing a kernel induced Hamiltonian flow as in Section 4.1 from the gradients of the infinite dimensional exponential family model (2.6), and approximate estimators (4.3), (4.5), we arrive at a gradient-free, adaptive MCMC algorithm: *Kernel Hamiltonian Monte Carlo* (Algorithm 1).

Computational efficiency and geometric ergodicity

KMC finite using (4.5) allows for online updates using the *full* Markov chain history $\{X_i\}_{i=1}^t$, and therefore is a more elegant solution than KMC lite, which has greater computational cost and requires sub-sampling the chain history. Due to the parametric nature of KMC finite, however, the tails of the estimator are not guaranteed to be flat. For example, the random Fourier feature embedding described below Proposition 4 contains periodic cosine functions, and therefore oscillates in the tails of (4.5), resulting in 'distracted' HMC proposals that are rarely accepted. As we will demonstrate in the experiments in Section 4.4, this problem does not appear when KMC finite is initialised in high-density regions, nor after burn-in.

In situations where information about the target density support is unknown, and during burn-in, we suggest to use the lite estimator (4.4), whose gradients decay outside of the training data. As a result, KMC lite is guaranteed to fall back to a random walk Metropolis (c.f. Section 2.3) in unexplored regions, inheriting its convergence properties, and smoothly transitions to HMC-like proposals as the MCMC chain grows. We have made this observations already in Chapter 3, where the kernel adaptive Metropolis-Hastings proposal's covariance matrix falls back to the scaled identity in unexplored regions. For the present case, which is more complex due to the nature of the used Hamiltonian dynamics, the following result establishes this formally, a proof can be found in Section 4.5.2.

Theorem 1. *Assume $d = 1$, $\pi(x)$ has log-concave tails, the regularity conditions of [96, Theorem 2.2] (implying π-irreducibility and smallness of compact sets), that MCMC adaptation stops after a fixed time, and a fixed number L of ϵ-leapfrog steps. If $\limsup_{\|x\|_2 \to \infty} \|\nabla f(x)\|_2 = 0$, and $\exists M : \forall x : \|\nabla f(x)\|_2 \leq M$, then KMC lite is geometrically ergodic from π-almost any starting point.*

For example, when using the Gaussian kernel with $\nabla f(x) = \sum_{i=1}^n \alpha_i \nabla k(x, X_i)$, it is straight-forward to verify that ∇f is both bounded and decays at infinity; a consequence of continuity and the fact that

$$\limsup_{\|x\|_2 \to \infty} (k(x, X)) = \limsup_{\|x\|_2 \to \infty} \left(\exp \left(-\sigma^{-1} \|x - X\|_2^2 \right) \right) = 0.$$

Vanishing adaptation and a full stop

MCMC algorithms that use the trajectory history for constructing proposals might yield divergent Markov chains. For the random walk proposals in Chapter 3 and Section 2.3, we used the idea of 'vanishing adaptation' to avoid such biases. To our knowledge, in the present case of learning gradients from the Markov chain history, however, it is not necessarily true that such diminishing adaptation schemes satisfy the conditions required for convergence, e.g. [94]. Therefore, we here take the conservative approach of using a diminishing adaptation for a fixed number of iterations, and then stop adaptation altogether. I.e. let $\{a_t\}_{i=0}^\infty$ be a schedule of decaying probabilities such that $\exists M : \forall t > M : a_t = 0$. We update the density gradient estimate according to this schedule in Algorithm 1. This schedule side-steps any potential problems with MCMC convergence.

The above adaptation schedule raises the natural question whether stopping to adapt 'too early' harms the efficiency of KMC – potentially resulting in a MCMC algorithm that is less efficient than an adaptive random walk scheme? This is in fact not the case: in Theorem 1 we established that KMC (lite) falls back to a random walk in unexplored regions (just as KAMH from Chapter 3 did), therefore inheriting efficiency guarantees.

In practice, we did not observe divergent MCMC chains, even when $\lim_{t\to\infty} a_t = 0$ and $\sum_{i=0}^{\infty} a_t = \infty$, which hints at the fact that theoretical work is needed to establish convergence conditions of adaptive MCMC with learned gradients. Marshall and Roberts [81] established theory for preconditioned Langevin-based methods adaptive MCMC methods, where the preconditioning matrix learned from past samples.

Free Parameters

KMC has two free parameters: the Gaussian kernel bandwidth σ, and the regularisation parameter λ. As KMC's performance depends on the quality of the approximate infinite dimensional exponential family model in (4.3) or (4.5), a principled approach is to use the score matching objective function in (2.8) to choose σ, λ pairs via cross-validation. The search can be guided by gradient-free black-box optimisation methods such as Bayesian optimisation or evolutionary approaches [62, 113].

This is an improvement to the earlier kernel adaptive Metropolis-Hastings from Chapter 3, where we did not address parameter choice due to the lack of an objective function that is optimised.

4.4 Experiments

We start by quantifying performance of KMC finite on synthetic targets; similar results can be produced with the lite version

Algorithm 1 Kernel Hamiltonian Monte Carlo – Pseudo-code

```
Input Target (possibly noisy estimator) $\hat{\pi}$, adaptation schedule $a_t$, HMC
parameters, size of basis $m$ or sub-sample size $n$.
At iteration $t+1$, current state $X_t$, history $\{X_i\}_{i=1}^t$, perform (1-4) with probability $a_t$

KMC lite:
  1. Update sub-sample of $\{X_i\}_{i=1}^t$
  2. Re-compute $C, b$ from Prop. 3
  3. Solve $\hat{\alpha}_\lambda = -\frac{\sigma}{2}(C + \lambda I)^{-1} b$
  4. $\nabla f(x) \leftarrow \sum_{i=1}^n \alpha_i \nabla k(x, X_i)$

KMC finite:
  1. Update to $C, b$ from Prop. 4
  2. Perform rank-$d$ update to $C^{-1}$
  3. Update $\hat{\theta}_\lambda = (C + \lambda I)^{-1} b$
  4. $\nabla f(x) \leftarrow [\nabla \phi_x]^\top \hat{\theta}$

  5. Propose $(p', X^*)$ with kernel induced Hamiltonian flow, using $\nabla_x U = \nabla_x f$
  6. Perform Metropolis step using $\hat{\pi}$: accept $X_{t+1} \leftarrow X^*$ w.p. (4.2) and
     reject $X_{t+1} \leftarrow X_t$ otherwise. If $\hat{\pi}$ is noisy and $X^*$ was accepted, store
     above $\hat{\pi}(X^*)$ for evaluating (4.2) in the next iteration.
```

4.4.1 KMC finite: stability of trajectories in high dimensions

In order to quantify efficiency in growing dimensions, we study hypothetical acceptance rates along proposal trajectories from high-density points; this is solely proposal based and there is no Markov chain produced yet. Consider a challenging Gaussian target: we sample the diagonal entries of the covariance matrix from a `Gamma(1,1)` distribution and then multiply the matrix with a uniformly sampled random rotation matrix [7]. The resulting target is challenging to estimate, as its length-scales are substantially different across principal components. As a single Gaussian kernel is not able to efficiently represent such scaling families, we use a rational quadratic kernel for the gradient estimation,

$$k(x,y) = \int \exp\left(-\frac{\|x-y\|_2^2}{2\ell^2}\right) p(\ell) \mathrm{d}\ell,$$

which an infinite sum of standard Gaussian kernels, weighted by a Gamma distribution $\mathcal{P}(\ell)$ over length scales ℓ.

Figure 4.2 shows the average acceptance over 100 independent trials as a function of the number of (ground truth) samples and basis functions, which are set to be equal $n = m$, and of dimension d. In low to moderate dimensions, gradients of the finite estimator lead to acceptance rates comparable to plain HMC.

On targets with more 'regular' smoothness, the estimator performs significantly better. We reproduce the experiment in Figure 4.2 on an isotropic Gaussian target. As length-scales across all principal components are equal, this is a significantly less challenging target to estimate gradients for; though still useful as a benchmark representing very smooth targets. We use a standard Gaussian kernel and the same experimental protocol as for Figure 4.2. The estimator performs well up to $d \approx 100$ with less variance, Figure 4.3.

4.4.2 KMC finite: mixing on a synthetic example

We next demonstrate that KMC's performance approaches that of HMC as it sees more data. We compare KMC, plain HMC [86], an isotropic random walk (RW), and KAMH from Chapter 3 on the 8-dimensional non-linear banana-shaped target that was introduced in Section 3.3.2. We here only quantify mixing *after* a sufficient burn-in (burn-in speed is included in next example). We quantify performance on estimating the target's mean, which is exactly $\mathbf{0}$.

We tune the scaling of KAMH and RW to achieve 23% acceptance using the Robbins-Monro recursion (2.17), as outlined in Section 2.3. We set HMC parameters to achieve 80% acceptance and then use the *same* parameters for KMC in order to ensure comparability. We run all samplers for 2000+200 iterations from a random start point, discard the burn-in and compute acceptance rates, the norm of the empirical mean $\|\hat{\mathbb{E}}[x]\|$, and the minimum effective sample size (ESS) across dimensions. For KAMH and

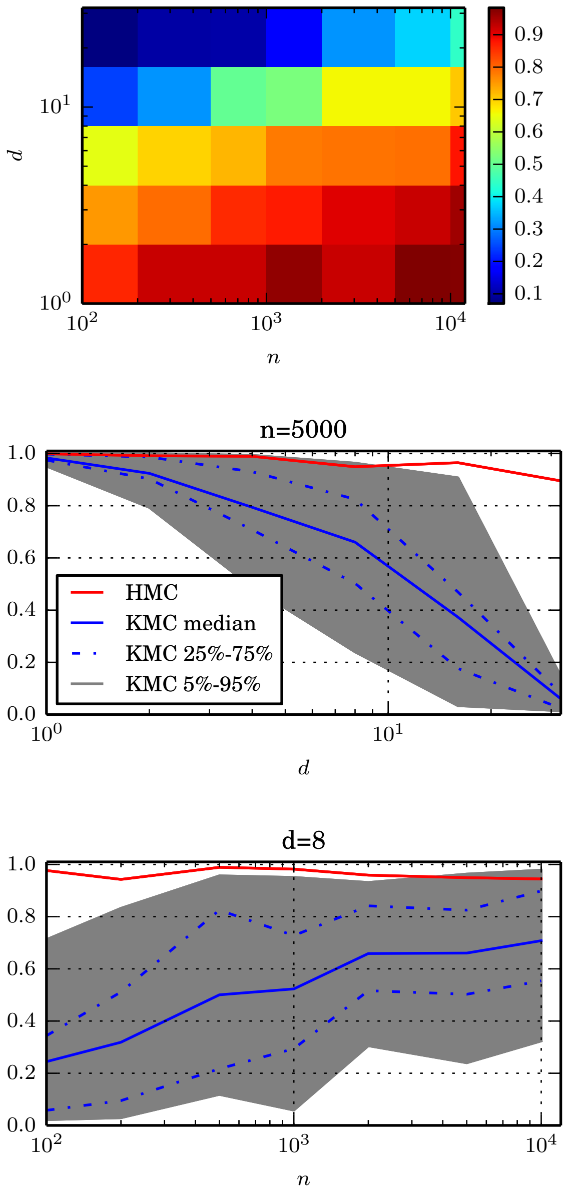


Figure 4.2: Hypothetical acceptance probability of KMC finite on a challenging target in growing dimensions. **Top:** As a function of $n = m$ (x-axis) and d (y-axis). **Middle/bottom:** Slices through first plot with error bars for fixed $n = m$ and as a function of d, and for fixed d as a function of $n = m$.

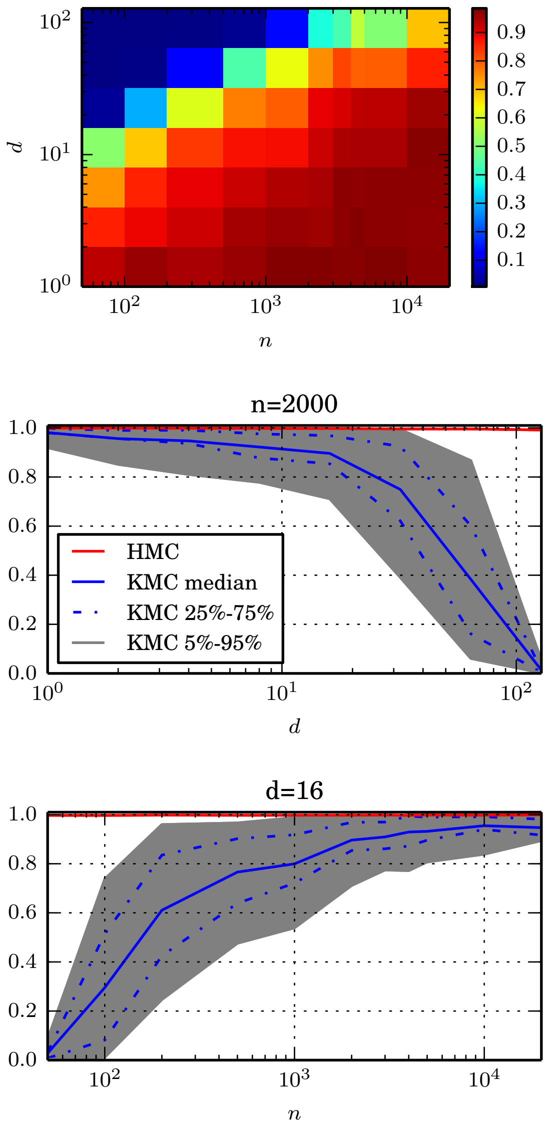


Figure 4.3: Acceptance probability of kernel induced Hamiltonian flow for a standard Gaussian in high dimensions for an isotropic Gaussian. **Top:** As a function of $n = m$ (x-axis) and d (y-axis). **Middle/bottom**: Slices through first plot with error bars for a fixed $n = m$ and as a function in d, and for a fixed d as a function of $n = m$.

KMC, we repeat the experiment for an increasing number of samples and basis functions $m = n$ to learn the density from. Figure 4.4 shows the results as a function of $m = n$. KMC clearly outperforms RW and KAMH, and eventually achieves performance close to HMC as $n = m$ grows.

4.4.3 KMC lite: pseudo-marginal MCMC for GPs

We next apply KMC to sample from the marginal posterior over hyperparameters of a Gaussian process classification model on the UCI glass dataset [9], as set up in Section 3.3.1. Classical HMC cannot be used for this problem, due to the intractability of the marginal likelihood and its gradients. We consider classification of window against non-window glass in the UCI Glass dataset, which induces a posterior that has a non-linear shape, see Figure 3.4. Figure 4.5 shows the posterior's pairwise marginals and corresponding kernel induced Hamiltonian dynamics.

Our experimental protocol mostly follows Section 3.3.1, though we only use 6000 MCMC samples. Since the ground truth for the hyperparameter posterior is not available, we initially run multiple hand-tuned standard Metropolis-Hastings chains for 500,000 iterations (with a 100,000 burn-in), keep every 1000-th sample in each of the chains, and combine them. The resulting samples are used as a benchmark, to evaluate the performance all algorithms. Once again, we use the MMD to assess convergence in the first three moments, c.f. Section 2.1, between each sampler output and the benchmark sample.

KMC randomly uses between 1 and 10 leapfrog steps of a size chosen uniformly in [0.01,0.1], a standard Gaussian momentum, and a kernel tuned by cross-validation, see below. We do not extensively tune the HMC parameters of KMC as the described settings are sufficient. Both KMC and KAMH use 1000 samples from the chain history.

Figure 4.6 (top) shows that KMC's burn-in contains a short 'exploration phase' where produced estimates are bad, due to it falling back to a random walk in unexplored regions, c.f. Theorem 1. From around

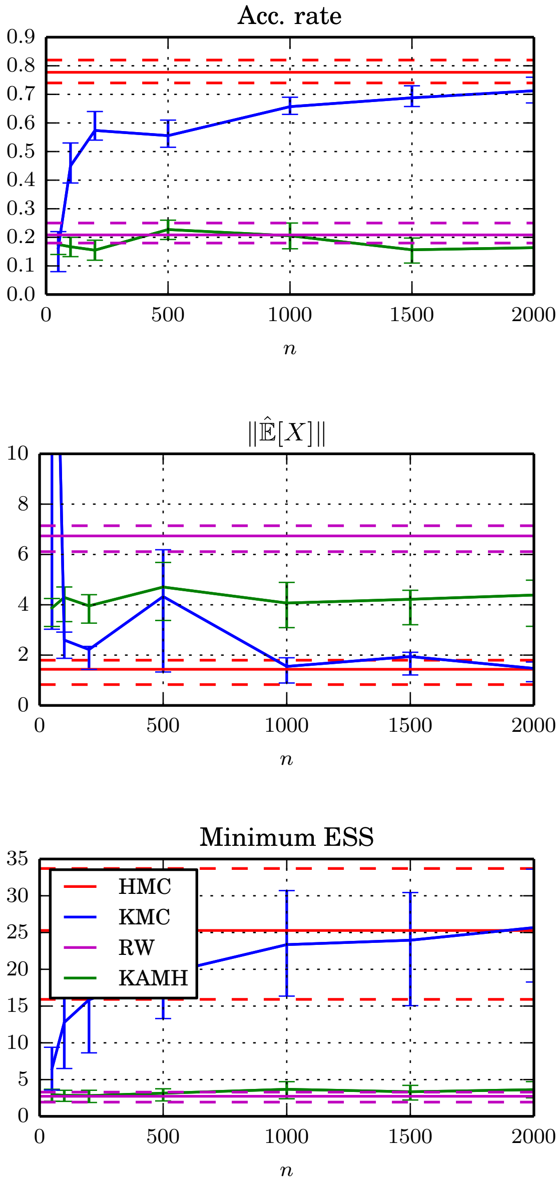


Figure 4.4: Results for the 8-dimensional synthetic Banana. As the amount of observed data increases, KMC performance approaches HMC – outperforming KAMH and RW. 80% error bars over 30 runs.

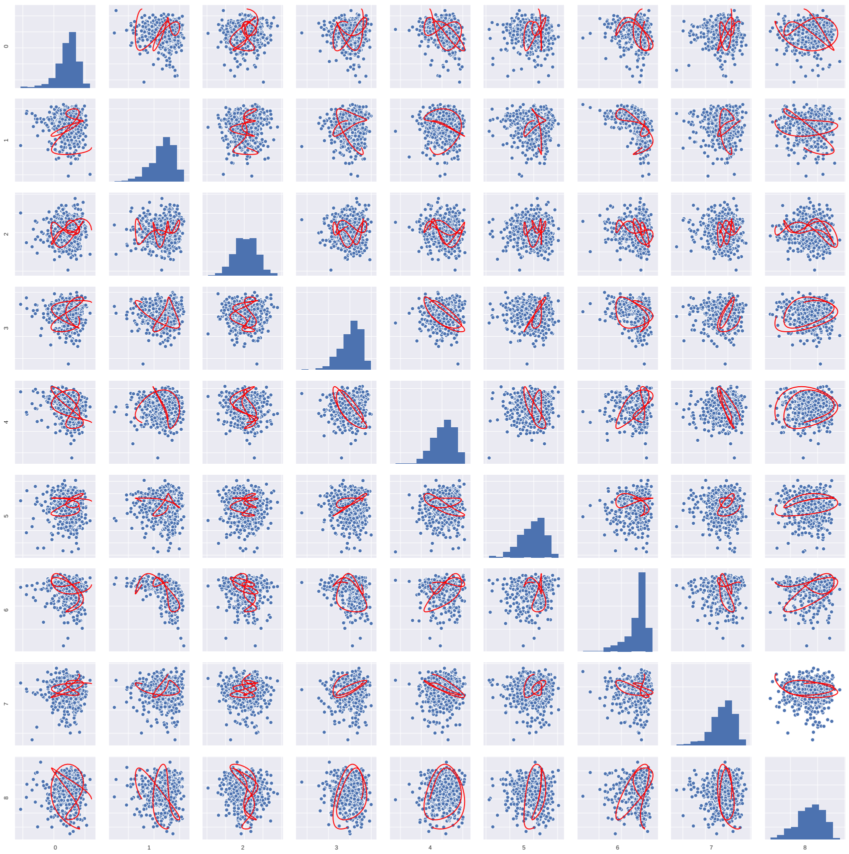


Figure 4.5: Pairwise marginals of the posterior over hyper-parameters of a Gaussian process classification model on the UCI glass dataset; along with kernel induced Hamiltonian dynamics that cover wide parts of the space. The non-linear marginal of Figure 3.4 is included.

500 iterations, however, KMC clearly outperforms both RW and the earlier state-of-the-art KAMH. These results are backed by the minimum ESS (not plotted), which is around 415 for KMC and is around 35 and 25 for KAMH and RW, respectively. Note that all samplers effectively stop improving from 3000 iterations – indicating a burn-in bias. All samplers took 1h time, with most time spent estimating the marginal likelihood.

Cross-validation

Kernel parameters are tuned using a black box Bayesian optimisation package and the median heuristic, for KMC and KAMH respectively. The Bayesian optimisation uses standard parameters and is stopped after 15 iterations, where each trial is done via a 5-fold cross-validation of the empirical score matching objective (2.8). We learn parameters after MCMC 500 iterations, and then re-learn after 2000. We tried re-learning parameters after more iterations, but this did not lead to significant changes. The costs for this are neglectable in the context of pseudo-marginal MCMC as estimating the marginal likelihood takes significantly more time than generating the KMC proposal.

4.4.4 KMC lite: reduced simulations and no additional bias in ABC

We now apply KMC in the context of approximate Bayesian computation (ABC), which often is employed when the data likelihood is intractable but can be obtained by simulation [108]. ABC-MCMC [80] targets an approximate posterior by constructing an unbiased Monte Carlo estimator of the approximate likelihood. As each such evaluation requires expensive simulations from the likelihood, the goal of all ABC methods is to reduce the number of such simulations. Accordingly, Hamiltonian ABC was recently proposed [82], combining the synthetic likelihood approach [132] with gradients based on stochastic finite differences, and using the stochastic gradient HMC framework by Chen et al. [31]. We remark that

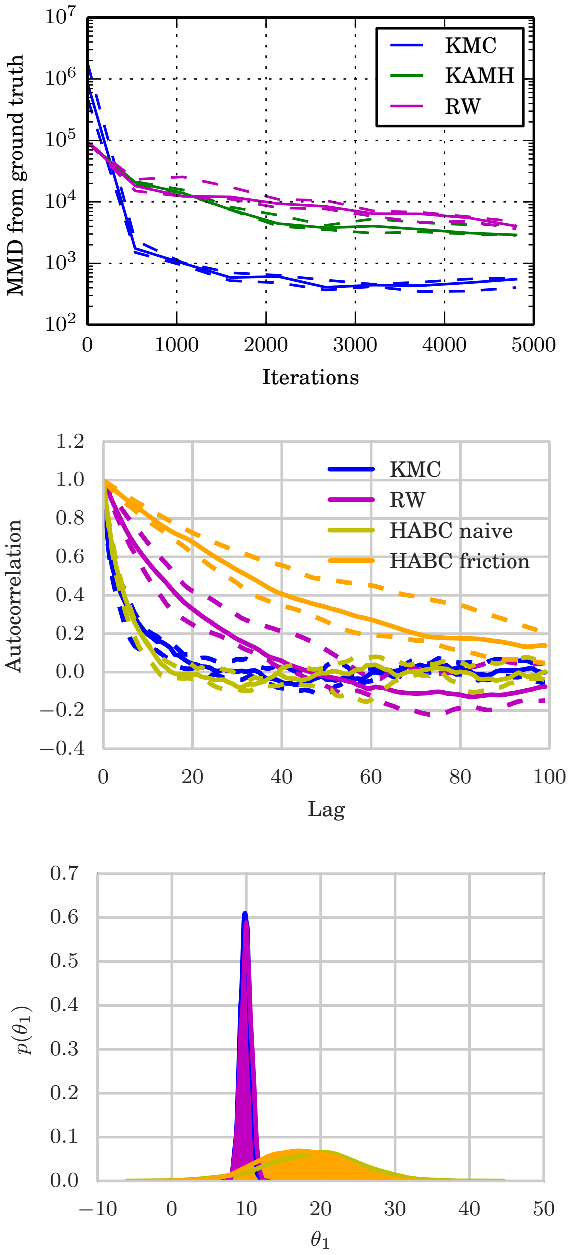


Figure 4.6: Top: Results for 9-dimensional marginal posterior over length scales of a GP classification model applied to the UCI Glass dataset. The plots show convergence (no burn-in discarded) of all mixed moments up to order 3 (lower MMD is better). **Middle/bottom:** ABC-MCMC auto-correlation and marginal θ_1 posterior for a 10-dimensional skew normal likelihood. While KMC mixes as well as HABC, it does not suffer from any bias (overlaps with RW, while HABC is significantly different) and requires fewer simulations per proposal. We also show performance of HABC with added friction, which has a severely negative impact on mixing.

this requires to simulate from the likelihood in *every* leapfrog step, and that the additional bias from the Gaussian likelihood approximation can be problematic. In contrast, KMC does not require simulations to construct a proposal, but rather 'invests' simulations into an accept/reject step (4.2) that ensures convergence to the *original* ABC target.

Target and experimental protocol

We generalise the above example to a 10-dimensional skew-normal distribution

$$p(y|\theta) = 2\mathcal{N}(y \mid \theta, I)\,\Phi\left(\alpha^\top y\right)$$

with $\theta = \alpha = \mathbf{1} \cdot 10 \in \mathbb{R}^{10}$ and Φ being the cumulative distribution function of the normal distribution. In each iteration of KMC, the likelihood is estimated via simulating $n_{\text{lik}} = 10$ samples from the above likelihood. We use the mean of all samples as summary statistic, and a Gaussian ABC similarity kernel with a bandwidth $\epsilon = 0.55$ [82, 108]. It is clear that sampling from a log-normal (skewed) and using the sample mean in a Gaussian likelihood leads to systematic upwards bias in the ABC scheme.

In terms of HMC parameters, both KMC and HABC use a standard Gaussian momentum, a uniformly random step-size in $[0.01, 0.1]$ and $L = 50$ leapfrog steps. HABC is used with the suggested 'sticky random numbers' [82, Section 4.4], i.e. we use the same seed for all simulations along a single proposal trajectory. Both algorithms are run for $200 + 5000$ MCMC iterations. KMC then attempts to re-learn smoothness parameters, and stops adaptation. Burn-in samples are discarded when quantifying performance of all algorithms.

Results: bias, friction, mixing, and number of simulations

Figure 4.6 (middle/bottom) compares performance of RW, HABC (sticky random numbers and SPAS, [82, Sections 4.3, 4.4]). KMC mixes as well as HABC, but HABC suffers from a severe bias.

HABC is used in its 'stochastic gradient' form [31] and has a 'friction' parameter that we estimate using a running average of the global covariance of all SPAS gradient evaluations, [82, Equation 21]. We run HABC with both the friction term included and removed, where we found that adding friction has severely negative impact on mixing, where not adding friction results in a wider posterior (with the same bias), c.f. Figure 4.6.

Due to the gradient estimation in every of the $L = 50$ leap-frog steps, *every* MCMC proposal for HABC requires $2L = 100$ simulations to be generated. In contrast, KMC only requires a single simulation, for evaluating the accept/reject probability (4.2).

We leave studying the exact trade-offs of KMC's learning phase and its ability to mix well as compared to HABC to future work.

4.5 Details for lite estimator

4.5.1 Proof of Proposition 3

The proof below extends the model in [64, Section 4.1]. We assume that the model log-density (2.6) takes the form in Proposition 3, then directly implement score functions (2.8), from which we derive an empirical score matching objective as a system of linear equations.

Proof. As assumed, the log un-normalised density takes the form

$$f(x) = \sum_{i=1}^{n} \alpha_i k(X_i, x)$$

where $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the Gaussian kernel in the form

$$k(X_i, x) = \exp\left(-\frac{1}{\sigma}\|X_i - x\|^2\right) = \exp\left(-\frac{1}{\sigma}\sum_{\ell=1}^{d}((X_i)_\ell - x_\ell)^2\right).$$

The score functions for (2.8) are then given by

$$\frac{\partial f}{\partial x_\ell} = \frac{2}{\sigma}\sum_{i=1}^{n}\alpha_i((X_i)_\ell - x_\ell)\exp\left(-\frac{\|X_i - x\|^2}{\sigma}\right),$$

and

$$
\begin{aligned}
\frac{\partial^2 f}{\partial^2 x_\ell} &= -\frac{2}{\sigma}\sum_{i=1}^{n}\alpha_i \exp\left(-\frac{\|X_i - x\|^2}{\sigma}\right) \\
&+ \left(\frac{2}{\sigma}\right)^2 \sum_{i=1}^{n}\alpha_i((X_i)_\ell - x_\ell)^2 \exp\left(-\frac{\|X_i - x\|^2}{\sigma}\right) \\
&= \frac{2}{\sigma}\sum_{i=1}^{n}\alpha_i \exp\left(-\frac{\|X_i - x\|^2}{\sigma}\right)\left[-1 + \frac{2}{\sigma}((X_i)_\ell - x_\ell)^2\right].
\end{aligned}
$$

Substituting this into the empirical version of (2.8) yields

$$
\begin{aligned}
\hat{J}(\alpha) &= \frac{1}{n}\sum_{i=1}^{n}\sum_{\ell=1}^{d}\left[\partial_\ell \psi_\ell(X_i;\alpha) + \frac{1}{2}\psi_\ell(X_i;\alpha)^2\right] \\
&= \frac{2}{n\sigma}\sum_{\ell=1}^{d}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i \exp\left(-\frac{\|X_i - x_j\|^2}{\sigma}\right)\left[-1 + \frac{2}{\sigma}((X_i)_\ell - x_{j\ell})^2\right] \\
&\quad + \frac{2}{n\sigma^2}\sum_{\ell=1}^{d}\sum_{i=1}^{n}\left[\sum_{j=1}^{n}\alpha_j(x_{j\ell} - (X_i)_\ell)\exp\left(-\frac{\|X_i - x_j\|^2}{\sigma}\right)\right]^2.
\end{aligned}
$$

We now rewrite $\hat{J}(\alpha)$ in matrix form. The expression for the term $\hat{J}(\alpha)$ being optimised is the sum of two terms.

First Term:

$$
\sum_{\ell=1}^{d}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i \exp\left(-\frac{\|X_i - x_j\|^2}{\sigma}\right)\left[-1 + \frac{2}{\sigma}((X_i)_\ell - x_{j\ell})^2\right]
$$

We only need to compute

$$
\begin{aligned}
&\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i \exp\left(-\frac{\|X_i - x_j\|^2}{\sigma}\right)((X_i)_\ell - x_{j\ell})^2 \\
=&\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i \exp\left(-\frac{\|X_i - x_j\|^2}{\sigma}\right)\left((X_i)_\ell^2 + x_{j\ell}^2 - 2(X_i)_\ell x_{j\ell}\right).
\end{aligned}
$$

Define

$$
x_\ell := \left[\begin{array}{ccc} (X_1)_\ell & \ldots & (X_m)_\ell \end{array}\right]^\top .
$$

The final term may be computed with the right ordering of operations,

$$-2(\alpha \odot x_\ell)^\top K x_\ell,$$

where $\alpha \odot x_\ell$ is the entry-wise product. The remaining terms are sums with constant row or column terms. Define $s_\ell := x_\ell \odot x_\ell$ with components $s_{i\ell} = (X_i)_\ell^2$. Then

$$\sum_{i=1}^n \sum_{j=1}^n \alpha_i k_{ij} s_{j\ell} = \alpha^\top K s_\ell.$$

Likewise

$$\sum_{i=1}^n \sum_{j=1}^n \alpha_i (X_i)_\ell^2 k_{ij} = (\alpha \odot s_\ell)^\top K \mathbf{1}.$$

Second Term: Considering only the ℓ-th dimension, this is

$$\sum_{i=1}^n \left[\sum_{j=1}^n \alpha_j (x_{j\ell} - (X_i)_\ell) \exp\left(-\frac{\|X_i - x_j\|^2}{\sigma} \right) \right]^2 .$$

In matrix notation, the inner sum is a column vector,

$$K(\alpha \odot x_\ell) - (K\alpha) \odot x_\ell.$$

We take the entry-wise square and sum the resulting vector. Denote by $D_x := \operatorname{diag}(x)$, then the following two relations hold

$$K(\alpha \odot x) = K D_x \alpha,$$

$$(K\alpha) \odot x = D_x K \alpha.$$

This means that $J(\alpha)$ as defined previously,

$$\begin{aligned}\hat{J}(\alpha) &= \frac{2}{n\sigma}\sum_{\ell=1}^{d}\left[\frac{2}{\sigma}\left[\alpha^\top K s_\ell + (\alpha\odot s_\ell)^\top K\mathbf{1} - 2(\alpha\odot x_\ell)^\top K x_\ell\right] - \alpha^\top K\mathbf{1}\right] \\ &+ \frac{2}{n\sigma^2}\sum_{\ell=1}^{d}\left[(\alpha\odot x_\ell)^\top K - x_\ell^\top \odot (\alpha^\top K)\right]\left[K(\alpha\odot x_\ell) - (K\alpha)\odot x_\ell\right],\end{aligned}$$

can be rewritten as

$$\begin{aligned}\hat{J}(\alpha) &= \frac{2}{n\sigma}\alpha^\top \sum_{\ell=1}^{d}\left[\frac{2}{\sigma}(Ks_\ell + D_{s_\ell}K\mathbf{1} - 2D_{x_\ell}Kx_\ell) - K\mathbf{1}\right] \\ &+ \frac{2}{n\sigma^2}\alpha^\top\left(\sum_{\ell=1}^{d}\left[D_{x_\ell}K - KD_{x_\ell}\right]\left[KD_{x_\ell} - D_{x_\ell}K\right]\right)\alpha \\ &= \frac{2}{n\sigma}\alpha^\top b + \frac{2}{n\sigma^2}\alpha^\top C\alpha,\end{aligned}$$

where

$$\begin{aligned}b &= \sum_{\ell=1}^{d}\left(\frac{2}{\sigma}(Ks_\ell + D_{s_\ell}K\mathbf{1} - 2D_{x_\ell}Kx_\ell) - K\mathbf{1}\right) \in \mathbb{R}^n, \\ C &= \sum_{\ell=1}^{d}\left[D_{x_\ell}K - KD_{x_\ell}\right]\left[KD_{x_\ell} - D_{x_\ell}K\right] \in \mathbb{R}^{n\times n}.\end{aligned}$$

Assuming C is invertible, this is minimised by

$$\hat{\alpha} = -\frac{\sigma}{2}C^{-1}b.$$

□

Similar to Sriperumbudur et al. [117], we add a term $\lambda\|\alpha\|_2^2$ to C, in order to control the L_2 norm[4] of the α coefficients. This results a numerically more stable solution. The corresponding minimised loss is $J(\alpha) + \frac{1}{2}n\sigma^2\lambda\|\alpha\|_2^2$.

[4] Sriperumbudur et al. [117] used the RKHS norm to regularise.

4.5.2 Proof for ergodicity of KMC lite, Theorem 1

Notation Denote by $\alpha(x_t, x^*(p'))$ the probability of accepting a (p', x^*) proposal at state x_t. Let $a \wedge b = \min(a, b)$. Define $c(x^{(0)}) := L\epsilon^2 \nabla \log \pi(x^{(0)})/2 + \epsilon^2 \sum_{i=1}^{L-1}(L-i)\nabla \log \pi(x^{(i\epsilon)})$ and $d(x^{(0)}) := \epsilon(\nabla f(x^{(0)}) + \nabla f(x^{(L\epsilon)}))/2 + \epsilon \sum_{i=1}^{L-1} \nabla f(x^{(i\epsilon)})$, where $x^{(i\epsilon)}$ is the i-th point of the leapfrog integration from $x = x^{(0)}$.

Proof. We assumed $\pi(x)$ is log-concave in the tails, meaning $\exists x_U > 0$ s.t. for $x^* > x_t > x_U$, we have $\pi(x^*)/\pi(x_t) \leq e^{-\alpha_1(|x^*|-|x_t|)}$ and for $x_t > x^* > x_U$, we have $\pi(x^*)/\pi(x_t) \geq e^{-\alpha_1(|x^*|-|x_t|)}$, and a similar condition holds in the negative tail. Furthermore, we assumed fixed HMC parameters: L leapfrog steps of size ϵ, and w.l.o.g. the identity mass matrix I. Following [83, 96], it is sufficient to show

$$\limsup_{|x_t| \to \infty} \int \left[e^{s(|x^*(p')|-|x_t|)} - 1 \right] \alpha(x_t, x^*(p')) \mu(dp') < 0,$$

for some $s > 0$, where $\mu(\cdot)$ is a standard Gaussian measure. Denoting the integral $I_{-\infty}^{\infty}$, we split it into

$$I_{-\infty}^{-x_t^\delta} + I_{-x_t^\delta}^{x_t^\delta} + I_{x_t^\delta}^{\infty},$$

for some $\delta \in (0,1)$. We show that the first and third terms decay to zero whilst the second remains strictly negative as $x_t \to \infty$ (a similar argument holds as $x_t \to -\infty$). We detail the case $\nabla f(x) \uparrow 0$ as $x \to \infty$ here, the other is analogous. Taking $I_{-x_t^\delta}^{x_t^\delta}$, we can choose an x_t large enough that $x_t - C - L\epsilon x_t^\delta > x_U$, $-\gamma_1 < c(x_t - x_t^\delta) < 0$ and $-\gamma_2 < d(x_t - x_t^\delta) < 0$. So for $p' \in (0, x_t^\delta)$ we have

$$L\epsilon p' > x^* - x_t > L\epsilon p' - \gamma_1 \implies e^{-\alpha_1(-\gamma_1 + L\epsilon p')} \geq e^{-\alpha_1(x^* - x_t)} \geq \pi(x^*)/\pi(x_t),$$

where the last inequality comes from the log-concave tails assumption. For

$p' \in (\gamma_2^2/2, x_t^\delta)$

$$\alpha(x_t, x^*) \leq 1 \wedge \frac{\pi(x^*)}{\pi(x_t)} \exp\left(p'\gamma_2/2 - \gamma_2^2/2\right) \leq 1 \wedge \exp\left(-\alpha_2 p' + \alpha_1\gamma_1 - \gamma_2^2/2\right),$$

where x_t is large enough that $\alpha_2 = \alpha_1 L\epsilon - \gamma_2/2 > 0$. Similarly for $p' \in (\gamma_1/L\epsilon, x_t^\delta)$

$$e^{sL\epsilon p'} - 1 \geq e^{s(x^* - x_t)} - 1 \geq e^{s(L\epsilon p' - \gamma_1)} - 1 > 0.$$

Because γ_1 and γ_2 can be chosen to be arbitrarily small, then for large enough x_t we will have

$$\begin{aligned} 0 < I_0^{x_t^\delta} &\leq \int_{\gamma_1/L\epsilon}^{x_t^\delta} [e^{sL\epsilon p'} - 1] \exp\left(-\alpha_2 p' + \alpha_1\gamma_1 - \gamma_2^2/2\right) \mu(dp') + I_0^{\gamma_1/L\epsilon} \\ &= e^{c_1} \int_{\gamma_1/L\epsilon}^{x_t^\delta} [e^{s_2 p'} - 1] e^{-\alpha_2 p'} \mu(dp') + I_0^{\gamma_1/L\epsilon}, \end{aligned} \tag{4.6}$$

where $c_1 = \alpha_1\gamma_1 - \gamma_2^2/2 > 0$ for large enough x_t, as γ_1 and γ_2 are of the same order. Now turning to $p' \in (-x_t^\delta, 0)$, we can use an exact rearrangement of the same argument (noting that c_1 can be made arbitrarily small) to get

$$I_{-x_t^\delta}^0 \leq e^{c_1} \int_{\gamma_1/L\epsilon}^{x_t^\delta} [e^{-s_2 p'} - 1] \mu(dp') < 0. \tag{4.7}$$

Combining (4.6) and (4.7) and rearranging as in Mengersen and Tweedie [83, Theorem 3.2] shows that $I_{-x_t^\delta}^{x_t^\delta}$ is strictly negative in the limit if $s_2 = sL\epsilon$ is chosen small enough, as $I_0^{\gamma_2/L\epsilon}$ can also be made arbitrarily small.

For $I_{-\infty}^{-x_t^\delta}$ it suffices to note that the Gaussian tails of $\mu(\cdot)$ will dominate the exponential growth of $e^{s(|x^*(p')| - |x_t|)}$ meaning the integral can be made arbitrarily small by choosing large enough x_t, and the same argument holds for $I_{x_t^\delta}^{\infty}$. □

4.6 Details for finite estimator

4.6.1 Proof of Proposition 4

We assume the model log-density (2.6) takes the primal form in a finite dimensional feature space as in Proposition 4, then again directly implement score functions in (2.8) and minimise it via a linear solve.

Proof. As assumed the log un-normalised density takes the form

$$f(x) = \langle \theta, \phi_x \rangle_{\mathcal{H}_m} = \theta^\top \phi_x,$$

where $x \in \mathbb{R}^d$ is embedded into a finite dimensional feature space $\mathcal{H}_m = \mathbb{R}^m$ as $x \mapsto \phi_x$. The score functions in (2.8) then can be written as the simple linear form

$$\frac{\partial f}{\partial x_\ell} = \theta^\top \dot{\phi}_x^\ell \quad \text{and} \quad \frac{\partial^2 f}{\partial x_\ell^2} = \theta^\top \ddot{\phi}_x^\ell, \tag{4.8}$$

where we defined the m-dimensional feature vector derivatives $\dot{\phi}_x^\ell := \frac{\partial}{\partial x_\ell}\phi_x$ and $\ddot{\phi}_x^\ell := \frac{\partial^2}{\partial x_\ell^2}\phi_x$. Plugging those into the empirical score matching objective (2.8), we arrive at

$$\begin{aligned}\hat{J}(\theta) &= \frac{1}{n}\sum_{i=1}^n \sum_{\ell=1}^d \left[\theta^\top \ddot{\phi}_{X_i}^\ell + \frac{1}{2}\theta^\top \left(\dot{\phi}_{X_i}^\ell \left(\dot{\phi}_{X_i}^\ell\right)^\top\right)\theta\right] \\ &= \frac{1}{2}\theta^\top C\theta - \theta^\top b \end{aligned} \tag{4.9}$$

where

$$b := -\frac{1}{n}\sum_{i=1}^n \sum_{\ell=1}^d \ddot{\phi}_{X_i}^\ell \in \mathbb{R}^m \quad \text{and} \quad C := \frac{1}{n}\sum_{i=1}^n \sum_{\ell=1}^d \left(\dot{\phi}_{X_i}^\ell \left(\dot{\phi}_{X_i}^\ell\right)^\top\right) \in \mathbb{R}^{m\times m}. \tag{4.10}$$

Assuming that C is invertible (trivial for $n \geq m$), the objective is uniquely minimised by differentiating (4.9) wrt. θ, setting to zero, and solving for θ. This gives

$$\hat{\theta} := C^{-1} b. \tag{4.11}$$

□

Again, similar to Sriperumbudur et al. [117], we add a term $\frac{\lambda}{2}\|\theta\|_2^2$ to (4.9), in order to control the norm of the natural parameters $\theta \in \mathcal{H}^m$ and for numerical stability.

Next, we give an example for the approximate feature space $\mathcal{H}_m$. Note that the above approach can be combined with *any* set of finite dimensional approximate feature mappings ϕ_x.

4.6.2 Example: random Fourier features for the Gaussian kernel

We now combine the finite dimensional approximate infinite dimensional exponential family model with the 'random Fourier features' by Rahimi and Recht [89]. Assume a translation invariant kernel $k(x,y) = \tilde{k}(x-y)$. Bochner's theorem gives the representation

$$k(x,y) = \tilde{k}(x-y) = \int_{\mathbb{R}^d} \exp\left(i\omega^\top(x-y)\right) \mathrm{d}\Gamma(\omega),$$

where $\Gamma(\omega)$ is the Fourier transform of the kernel. An approximate feature mapping for such kernels can be obtained via dropping imaginary terms and approximating the integral with Monte Carlo integration. This gives

$$\phi_x = \sqrt{\frac{2}{m}}\left[\cos(\omega_1^\top x + u_1),\ldots,\cos(\omega_m^\top x + u_m)\right],$$

with fixed random basis vector realisations that depend on the kernel via its Fourier transform $\Gamma(\omega)$,

$$\omega_i \sim \Gamma(\omega),$$

and fixed random offset realisations

$$u_i \sim \texttt{Uniform}[0,2\pi],$$

for $i = 1 \dots m$. It is easy to see that this approximation is consistent for $m \to \infty$, i.e.

$$\mathbb{E}_{\omega,b}\left[\phi_x^\top \phi_y\right] = k(x,y).$$

See Rahimi and Recht [89] for details and a uniform convergence bound and Sriperumbudur and Szábo [116] for a more detailed analysis with tighter bounds. Note that it is possible to achieve logarithmic computational costs in d exploiting properties of Hadamard matrices [71].

The feature map derivatives (4.8) are given by

$$\begin{aligned}
\dot{\phi}_\xi^\ell &= \sqrt{\frac{2}{m}}\frac{\partial}{\partial\xi_\ell}\left[\cos(\omega_1^\top\xi + u_1),\dots,\cos(\omega_m^\top\xi + u_m)\right] \\
&= -\sqrt{\frac{2}{m}}\left[\sin(\omega_1^\top\xi + u_1)\omega_{1\ell},\dots,\sin(\omega_m^\top\xi + u_m)\omega_{m\ell}\right] \\
&= -\sqrt{\frac{2}{m}}\left[\sin(\omega_1^\top\xi + u_1),\dots,\sin(\omega_m^\top\xi + u_m)\right] \odot [\omega_{1\ell},\dots,\omega_{m\ell}],
\end{aligned}$$

where $\omega_{j\ell}$ is the ℓ-th component of ω_j, and

$$\begin{aligned}
\ddot{\phi}_\xi^\ell &:= -\sqrt{\frac{2}{m}}\frac{\partial}{\partial\xi_\ell}\left[\sin(\omega_1^\top\xi + u_1),\dots,\sin(\omega_m^\top\xi + u_m)\right] \odot [\omega_{1\ell},\dots,\omega_{m\ell}] \\
&= -\sqrt{\frac{2}{m}}\left[\cos(\omega_1^\top\xi + u_1),\dots,\cos(\omega_m^\top\xi + u_m)\right] \odot \left[\omega_{1\ell}^2,\dots,\omega_{m\ell}^2\right] \\
&= -\phi_\xi \odot \left[\omega_{1\ell}^2,\dots,\omega_{m\ell}^2\right],
\end{aligned}$$

where $\odot$ is the element-wise product. Consequently the gradient is given by

$$\nabla_\xi \phi_\xi = \begin{bmatrix} \dot{\phi}_\xi^1 \\ \vdots \\ \dot{\phi}_\xi^d \end{bmatrix} \in \mathbb{R}^{d\times m}.$$

As an example, the translation invariant Gaussian kernel and its Fourier transform are

$$k(x,y) = \exp\left(-\sigma^{-1}\|x-y\|_2^2\right) \quad \text{and} \quad \Gamma(\omega) = \mathcal{N}\left(\omega \middle| \mathbf{0}, \sigma^{-2}I_m\right).$$

4.6.3 Constant cost updates

A convenient property of the finite feature space approximation is that its primal representation of the solution allows to update (4.10) in an online fashion. When combined with MCMC, each new point X_{t+1} of the Markov chain history only adds a term of the form $-\sum_{\ell=1}^{d} \ddot{\phi}^{\ell}_{X_{t+1}} \in \mathbb{R}^m$ and $\sum_{\ell=1}^{d} \dot{\phi}^{\ell}_{X_{t+1}} (\dot{\phi}^{\ell}_{X_{t+1}})^{\top} \in \mathbb{R}^{m \times m}$ to the moving averages of b and C respectively. Consequently, at iteration t, rather than fully re-computing (4.11) at the cost of $\mathcal{O}(tdm^2 + m^3)$ for every new point, we can use rank-d updates to construct the minimiser of (4.9) from the solution of the previous iteration. Assume we have computed the sum of all moving average terms,

$$\bar{C}_t^{-1} := \left(\sum_{i=1}^{t} \sum_{\ell=1}^{d} \left(\dot{\phi}^{\ell}_{X_i} \left(\dot{\phi}^{\ell}_{X_i} \right)^{\top} \right) \right)^{-1}$$

from feature vectors derivatives $\ddot{\phi}^{\ell}_{X_i} \in \mathbb{R}^m$ of some set of points $\{X_i\}_{i=1}^{t}$, and subsequently receive receive a new point X_{t+1}. We can then write the inverse of the new sum as

$$\bar{C}_{t+1}^{-1} : = \left(\bar{C}_t + \sum_{\ell=1}^{d} \left(\dot{\phi}^{\ell}_{X_{t+1}} \left(\dot{\phi}^{\ell}_{X_{t+1}} \right)^{\top} \right) \right)^{-1}.$$

This is the inverse of the rank-d perturbed previous matrix $\bar{C}_t$. We can therefore construct this inverse using d successive applications of the Sherman-Morrison-Woodbury formula for rank-one updates, each using $\mathcal{O}(m^2)$ computation. Since $\bar{C}_t$ is positive definite[5], we can represent its inverse as a numerically much more stable Cholesky factorisation $\bar{C}_t = \bar{L}_t \bar{L}_t^{\top}$. It is also possible to perform cheap rank-d updates of such Cholesky fac-

[5] C is the empirical covariance of the feature derivatives $\dot{\phi}^{\ell}_{X_i}$.

tors[6]. Denote by $\bar{b}_t$ the sum of the moving average b. We solve (4.11) as

$$\hat{\theta} = C^{-1}b = \left(\frac{1}{t}\bar{C}_t\right)^{-1}\left(\frac{1}{t}\bar{b}_t\right) = \bar{C}_t^{-1}\bar{b}_t = \bar{L}_t^{-\top}\bar{L}_t^{-1}\bar{b}_t,$$

using cheap triangular back-substitution from $\bar{L}_t$, and never storing $\bar{C}_t^{-1}$ or $\bar{L}_t^{-1}$ explicitly.

Using such updates, the computational costs for updating the approximate infinite dimensional exponential family model in *every* iteration of the Markov chain are $\mathcal{O}(dm^2)$, which *constant in t*. We can therefore use *all* points in the history for constructing a proposal. See our implementation for further details.

Algorithmic Description

1. Update sums

$$\bar{b}_{t+1} = \bar{b}_t - \sum_{\ell=1}^{d} \ddot{\phi}^{\ell}_{X_{t+1}} \quad \text{and} \quad \bar{C}_{t+1} = \bar{C}_t + \frac{1}{2}\sum_{\ell=1}^{d} \dot{\phi}^{\ell}_{X_{t+1}}(\dot{\phi}^{\ell}_{X_{t+1}})^{\top}.$$

2. Perform rank-d update to obtain updated Cholesky factorisation $\bar{L}_{t+1}\bar{L}_{t+1}^{\top} = \bar{C}_{t+1}$.

3. Update approximate infinite dimensional exponential family parameters

$$\hat{\theta} = \bar{L}_{t+1}^{-\top}\bar{L}_{t+1}^{-1}\bar{b}_{t+1}.$$

[6] We use the open-source implementation provided at `https://github.com/jcrudy/choldate`

Chapter 5

Kernel sequential Monte Carlo

This chapter is based on collaborative work I. Schuster, H. Strathmann, B. Paige, and D. Sejdinovic. “Kernel Adaptive Sequential Monte Carlo”. In: *European conference on machine learning & principles and practice of knowledge discovery in databases*. Joint first two authors. 2017.

We propose kernel sequential Monte Carlo (KSMC), a framework for sampling from static target densities. KSMC is a family of sequential Monte Carlo algorithms that are based on building surrogate models of the current particle system in a reproducing kernel Hilbert space. We here focus on modelling non-linear covariance structure and gradients of the target. The surrogate’s geometry is adaptively updated and subsequently used to inform local proposals. Unlike in adaptive Markov chain Monte Carlo, continuous adaptation does not compromise convergence of the sampler. KSMC combines the strengths of sequential Monte Carlo and kernel methods: superior performance for multi-modal targets and the ability to estimate model evidence as compared to Markov chain Monte Carlo, and the surrogate’s ability to represent targets that exhibit high degrees of non-linearity. As KSMC does not require access to target gradients, it is particularly applicable on targets whose gradients are unavailable. We demonstrate the benefits of the the proposed methodology on a series of challenging synthetic and real-world examples.

Chapter outline

We have seen in Chapter 3 and Chapter 4, how modelling covariance structure and gradients can greatly improve the efficiency of Metropolis-Hastings based MCMC algorithms. In this chapter, we will generalise these ideas to the context of sequential Monte Carlo for static targets. They key observation that allows this combination is that many SMC algorithms consist of an MCMC move, so-called rejuvenation step. As it turns out, it is possible to model the structure of the current particle system to make such moves more efficient in the same way as for MCMC.

5.1 Sequential Monte Carlo

Sequential Monte Carlo algorithms [40, 105] approximate a target density π by iteratively targeting a sequence of incremental densities $\pi_0,\ldots,\pi_T$, with $\pi_T = \pi$. These incremental densities are typically defined such that the initial density π_0 is easy to sample from (e.g. the prior in a Bayesian model). Consecutive distributions π_t,π_{t+1} are 'close', in the sense that drawing samples from π_{t+1} given samples from π_t is easier than drawing samples from π_{t+1} directly. At each stage t, we approximate the target density π_t with a set of n samples $\mathbf{X}_t = \{X_t^i\}_{i=1}^n$, with associated importance weights $\mathbf{W}_t = \{W_t^i\}_{i=1}^n$, with

$$\hat{\pi}_t(X) = \sum_{i=1}^n W_t^i \delta_{X_t^i}(X) \tag{5.1}$$

where $\delta_{X_t^i}$ is a Dirac point mass on X_t^i. In a static SMC setting, in contrast to SMC as applied to state space models, each target density π_t is defined on the same space $\mathcal{X}$.

We initialise the algorithm by sampling an initial set of n samples $\mathbf{X}_0$ from the initial density π_0, with uniform importance weights. For each subsequent $t = 1,\ldots,T$, given a particle set $(\mathbf{X}_{t-1},\mathbf{W}_{t-1})$ approximating π_{t-1}, we construct a new particle set to represent π_t. This is a three-step process of re-weighting, re-sampling, and rejuvenation, also summarised in Algorithm 2:

1. We re-weight each particle relative to the new target density, setting

   $$\widetilde{W}_t^i = W_{t-1}^i \frac{\pi_t(X_{t-1}^i)}{\pi_{t-1}(X_{t-1}^i)}.$$

   Weighting the points in $\mathbf{X}_{t-1}$ by $\{\widetilde{W}_t^i\}_{i=1}^n$ yields an approximation to π_t in the same manner as in (5.1) — the new importance weights correct for the change from π_{t-1} to π_t.

2. We apply re-sampling, constructing an equally-weighted set of particles $\widetilde{\mathbf{X}}_t = \{\widetilde{X}_t^i\}_{i=1}^n$ by sampling with replacement from $\mathbf{X}_{t-1}$ with weights proportional to $\widetilde{W}_t^i$, [38]. Together, these samples form an approximation to π_t, where values from $\mathbf{X}_{t-1}$ with high weight under π_t have been duplicated and those with low weight under π_{t-1} have been discarded.

   This duplication of values, however, can lead to a sample impoverishment problem: many of the re-sampled values $\widetilde{X}_t^i$ may have identical values. This can be avoided by applying a so-called *rejuvenation* step after re-sampling [32], constructing an overall approximation $(\mathbf{X}_t, \mathbf{W}_t)$ to π_t with a diverse set of values of X_t^i.

3. The rejuvenation is based on a proposal $Q_t(\mathbf{X}_t|\widetilde{\mathbf{X}}_t)$. One traditional option is to use a Markov density Q_t as a proposal in a Metropolis-Hastings kernel which leaves π_t invariant: For each $\widetilde{X}_t^i$ in $\widetilde{\mathbf{X}}_t$, we propose a new value X_t^i from $Q_t(X_t^i|\widetilde{X}_t^i)$ and accept it according to a standard MH acceptance ratio targeting π_t. In this case, importance

Algorithm 2 Sequential Monte Carlo for Static Models

```
Input: Sequence of target densities $\pi_0,\ldots,\pi_T$ (where $\pi_T = \pi$), size of particle system $n$
Output: sets $\mathbf{X}_1,\ldots,\mathbf{X}_T$ and $\mathbf{W}_1,\ldots,\mathbf{W}_T$ of samples and accompanying weights
Initialise $\mathbf{X}_0$ to $n$ samples from $\pi_0$, and $\mathbf{W}_0$ to equal weights $1/n$
for $t = 1$ through $t = T$ do
    $\widetilde{\mathbf{W}}_t = \{W^i_{t-1}\pi_t(X^i_{t-1})/\pi_{t-1}(X^i_{t-1})\}^n_{i=1}$
    construct $\widetilde{\mathbf{X}}_t$ by re-sampling $(\mathbf{X}_{t-1}, \widetilde{\mathbf{W}}_t)$, resulting in $n$ copies of samples in $\mathbf{X}_{t-1}$
    construct or update proposal $Q_t$
    if using a MH transition kernel then
        Set $\mathbf{X}_t$ to
            $\{X^i_t \sim$ MH kernel with proposal $Q_t(\cdot|\widetilde{X}^i_t)\}^n_{i=1}$
        $\mathbf{W}_t = \{1/n\}^n_{i=1}$
    end if
end for
return $\mathbf{X}_1,\ldots,\mathbf{X}_T$ and $\mathbf{W}_1,\ldots,\mathbf{W}_T$
```

weights in $\mathbf{W}_t$ are set to uniform.

5.1.1 Construction of a target sequence

One possibility for constructing a sequence of distributions is the geometric likelihood bridge defined by

$$\pi_t \propto \pi_0^{1-\rho_t}\pi^{\rho_t},$$

for some initial distribution π_0, where $(\rho_t)_{t=1}^T$ is an increasing sequence satisfying $\rho_T = 1$. As a simple alternative, we can simply target the final distribution π at each iteration, i.e. with all $\pi_t = \pi$. This algorithm is known as *population Monte Carlo* [28].

5.1.2 Existing adaptive SMC algorithms

In SMC algorithms, we are free in choosing a proposal Q_t. In contrast to MCMC, it may be directly informed by the previous samples $\mathbf{X}_{t-1}$ and their weights $\mathbf{W}_{t-1}$, without compromising consistency of the produced estimates. The following two existing SMC algorithms are examples that we will extend to kernel-based alternatives.

Adaptive SMC

The adaptive SMC sampler (ASMC) studied by Fearnhead and Taylor [43] is based on continuously estimating the global covariance Σ_t of π_t, and updating a scaling parameter ν^2. This is done from the re-weighted particle system, which is subsequently moved through a Markov kernel. The proposal distribution used within the MH kernel at point X in Algorithm 2 is

$$Q_t(\cdot|X) = \mathcal{N}(\cdot|X, \nu^2\Sigma_t + \gamma^2 I),$$

which is extremely similar to (2.15) and (2.16). Indeed, the algorithm is essentially the previously discussed adaptive Metropolis, c.f. Section 2.3 and Andrieu and Thoms [6] and Haario et al. [60], embedded into the SMC context.

Gradient importance sampling

In addition to using the estimated covariance Σ_t of π as in ASMC, gradient importance sampling [102, GRIS] incorporates a drift term based on the log target gradient. For target gradient $\nabla \log \pi$ and previous sample X, the proposal distribution in Algorithm 2 is

$$Q_t(\cdot) = \mathcal{N}(\cdot|X + D(\nabla \log \pi(X)), \nu^2\Sigma_t),$$

for each individual particle X in the current (un-weighted) particle set. A typical choice for the drift function is $D(y) = \delta y$ with $0 < \delta < 1$. Rather than incorporating a MH step, the updated values are importance weighted — GRIS is an instance of a population Monte Carlo algorithm. In numerical experiments, GRIS compares favourably to its closest MCMC relatives [102].

5.1.3 Intractable likelihoods

In the the case where likelihoods are intractable, SMC is still a valid algorithm when likelihood values can be estimated without bias. This can be done using e.g. importance sampling or SMC (nested within SMC, hence SMC^2) [33, 125]. Just like in the pseudo-marginal MCMC case, a simple way to think about such nested estimation schemes is in terms of an extended sampling space that spans the actual parameters of interest as well as any nuisance variables. Once again, efficient gradient-based sampling schemes such as GRIS or HMC are unavailable. Current practice there is based on moving particles using random walk schemes solely.

5.1.4 Evidence estimation

An important issue in Bayesian model selection and averaging is that of estimating the normalizing constant, or *evidence*. The evidence is the marginal probability of the data under a model and can easily be estimated in SMC instantiations [40, 43] – as compared to MCMC. This enables routine computation of Bayes factors and posterior model probabilities. Under the assumption that the normalizing constant Z_0 of π_0 (the distribution that is used for initially setting up the particle system) is known, one can estimate the ratio of normalizing constants of any two consecutive targets by

$$\frac{Z_t}{Z_{t-1}} \approx \frac{1}{n}\sum_{j=1}^{n} W_{t,j}$$

for $W_{t,j} = \pi_t(X_{t-1,j})/\pi_{t-1}(X_{t-1,j})$ and thus an estimate for $Z = Z_T$ can be found recursively by

$$Z = Z_T \approx Z_0 \prod_{t=1}^{T} \frac{1}{n} \sum_{j} W_{t,j}$$

starting with known value Z_0. When the likelihood is intractable and importance weights therefore are noisy, evidence estimation remains consistent given that the likelihood estimates are unbiased [125, Lemma 3].

5.2 Kernel sequential Monte Carlo

We now develop a kernel sequential Monte Carlo framework, which is based on combining classical adaptive SMC with the proposals of kernel adaptive MCMC from Chapter 3 and Chapter 4. In general, once a kernel surrogate is fitted to past particle systems, we can use it in either of two ways: as proposals for MH rejuvenation steps inside SMC, or as importance densities in PMC.

5.2.1 Kernel covariance adaptive rejuvenation: KASMC

At time-step $t+1$, we target distribution π_{t+1}, based on a particle system approximating π_t. After re-weighting, the new system $\{(W_{t+1,i}, X_{t+1,i})\}_{i=1}^n$ is a weighted approximation to π_{t+1}. We here focus on the non-linear covariance surrogate, based on Chapter 3, which can be either fitted using the equally-weighted re-sampled values $\widetilde{\mathbf{X}}_t$, or the original particle set $\mathbf{X}_t$ with weights $\widetilde{\mathbf{W}}_t$. Denote by Q_{KAMH} the MCMC proposal distribution based on a kernel covariance model from Chapter 3, Proposition 2; at particle X_j, this is (using $\widetilde{\mathbf{X}}_t$)

$$Q_{\text{KAMH}}(\cdot|X_j) = \mathcal{N}(\cdot|X_j, \gamma^2 I + \nu^2 M_{\widetilde{\mathbf{X}}_t, X_j} C M_{\widetilde{\mathbf{X}}_t, X_j}^\top).$$

As described in Section 3.2.3, for a Gaussian kernel, the proposal at particle X_j locally aligns to the structure of the posterior at X_j. The SMC rejuvenation proposal for Algorithm 2 at X then is exactly Q_{KAMH}. As in KAMH, this results in covariance matrices for Gaussian proposals which locally align with the target, c.f. Figure 3.3, now taking the SMC particle weights into account. The resulting kernel adaptive SMC sampler (KASMC) inherits KAMH's ability to explore non-linear targets more efficiently than proposals based on estimating global covariance structure such as in Fearnhead and Taylor [43] and Haario et al. [61]. Figure 5.1 shows a simple illustration of a global (ASMC) and local proposal distribution (KASMC).

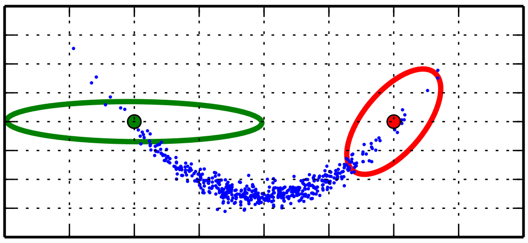


Figure 5.1: Proposal distributions around one of many SMC particles (blue) for each KASMC (red) and ASMC (green). KASMC proposals locally align to the target density while ASMC's global covariance estimate might result in poor MH rejuvenation moves.

Scaling parameters

To accomplish parameter tuning, we use the standard MCMC framework of stochastic approximation for tuning MCMC kernels, as outlined in Section 2.3. I.e. we tune the acceptance rate α_t towards an asymptotically optimal value of $\alpha^* = 0.234$, as also done in Chapter 3 and Chapter 4.

In SMC, after the MCMC rejuvenation step, an estimate of expected acceptance probability is obtained by averaging the acceptance probabilities for all MH proposals, $\hat{\alpha}_t = \frac{1}{t}\sum_{i=1}^{t} \alpha_t$, and we substitute this for α_t in the Robbins-Monro recursion (2.17). This strategy of approximating optimal scaling assumes that consecutive targets are close enough so that the acceptance rate when using ν_t to target π_t provides information about the expected acceptance rate when using ν_t with target π_{t+1}. As an alternative to this, one could treat ν_t as an auxiliary random variable and define a distribution over it designed to maximise expected utility, an approach taken in the adaptive SMC sampler [43].

5.2.2 Kernel induced gradients for importance densities: KGRIS

Another way to use kernel-based surrogates is for generating proposals which are corrected by importance sampling, i.e. in PMC, c.f. Section 5.1.1. In our second approach, a kernel surrogate is fitted to weighted particles, which were previously corrected via importance weights. As such, particles share the same underlying density across the PMC iterations. We here use the kernel gradient model from Chapter 4, in its finite dimensional approximation (KMC finite), c.f. Proposition 4.

Weighted random feature exponential family approximations

As in (4.5), the log density of the approximate estimator takes the simple form $f(x) = \theta^\top \phi_x$, where $\phi_x \in \mathbb{R}^m$ is an embedding of x into an m-dimensional feature space, and $\theta \in \mathbb{R}^m$ is estimated by $\hat{\theta} = C^{-1}b$ from samples x. Given a weighted particle system $\{(W_{t,i}, X_{t,i})\}_{i=1}^n$, then b, C are weighted averages of the form

$$
\begin{aligned}
b &:= -\frac{1}{\sum_{i=1}^n W_{t,i}} \sum_{i=1}^n W_{t,i} \sum_{\ell=1}^d \ddot{\phi}_x^\ell, \\
C &:= \frac{1}{\sum_{i=1}^n W_{t,i}} \sum_{i=1}^n W_{t,i} \sum_{\ell=1}^d \dot{\phi}_x^\ell \left(\dot{\phi}_x^\ell\right)^\top,
\end{aligned} \tag{5.2}
$$

with element-wise derivatives $\dot{\phi}_x^\ell := \frac{\partial}{\partial x_\ell}\phi_x$ and $\ddot{\phi}_x^\ell := \frac{\partial^2}{\partial x_\ell^2}\phi_x$. The only difference to Proposition 4 are the included sample weights. As described in Section 4.6.3, the estimator can be updated in an online fashion once the particle system changes. This is particularly for the present PMC context: as all particles share the same density, the model can accumulate information over time rather than being re-estimated in every iteration. Rather than simulating Hamiltonian dynamics to generate a proposal, we here follow the initial work of Schuster [102] and take single gradient steps, i.e. the Markov density at in Algorithm 2 at X_j is

$$
Q_t(\cdot|X_j) = \mathcal{N}(\cdot|X_j + \delta\nabla f(X_j), \nu^2\Sigma_t),
$$

with parameters $\delta > 0, \nu^2 > 0$, and using an online estimate of the global sample covariance Σ_t same as in Fearnhead and Taylor [43] and Haario et al. [60, 61], c.f. Section 2.3.

5.3 Experiments

We now evaluate empirical performance of KASMC on a simple non-linear target, on a multi-modal sensor network localisation problem, and in estimating Bayesian model evidence in a model with an intractable likelihood on a real-world dataset. The final experiment uses a challenging stochastic volatility model with S&P 500 data from Chopin et al. [33] to evaluate KGRIS.

For the KASMC experiments on static target distributions, we use the geometric likelihood bridge described in Section 5.1.1. The bandwidth parameter of the kernel surrogate models is set to the median distance between particles [54, 104]. We tune the free scaling parameter ν towards the asymptotically 'optimal' acceptance rate $\alpha^* = 0.234$, c.f. Section 5.2.1.

5.3.1 KASMC: Improved convergence on synthetic non-linear target

We begin by studying convergence of KASMC compared to existing algorithms on a simple benchmark example: the strongly twisted banana-shaped distribution in $d = 8$ dimensions, c.f. Section 3.3.2. We compare SMC algorithms using different rejuvenation MH steps: a static random walk Metropolis move (RWSMC) with fixed scaling $\nu = 2.38/\sqrt{d}$, ASMC, and KASMC using a Gaussian kernel. For the latter two algorithms, all particles are used to compute the proposal, and a fixed learning rate of $\lambda = 0.1$ is chosen to adapt scale parameters, c.f. Section 2.3. Starting with particles from a multivariate Gaussian $\mathcal{N}(0,50^2)$, we use a geometric bridge that reaches the target $\mathcal{B}(y;b = 0.1, v = 100)$ in 20 steps. We repeat the experiment over 30 runs. Figure 5.2 shows that KASMC achieves faster convergence of the first 3 moments, i.e. in MMD distance, c.f. Section 2.1, to a large benchmark sample.

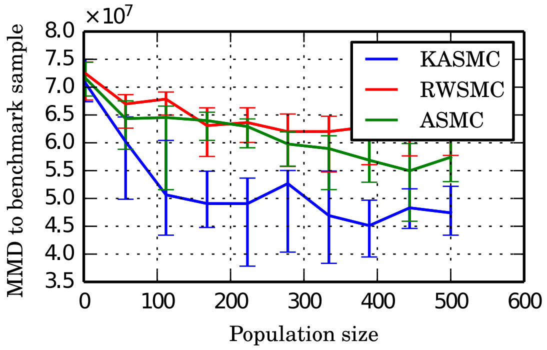


Figure 5.2: Improved convergence of all mixed moments up to order 3 of KASMC compared to using SMC with static or adaptive Metropolis-Hastings steps.

5.3.2 A multi-modal application: sensor network localisation

We next study performance of KASMC on a multi-modal target arising in a real-world application: inferring the locations of S sensors within a network, as discussed in [65, 70]. We here focus on the static case: assume a number of stationary sensors that measure distance to each other in a 2-dimensional space; a distance measurement is successful with a probability that decays exponentially in the squared distance, and the observation is missing otherwise. If distance is measured, it is corrupted by Gaussian noise. The posterior over the unknown sensor locations, induced by the spatial set-up, forms a highly constrained non-linear and multi-modal distribution.

Denote the S unknown sensor locations by $\{x_i\}_{i=1}^S \subseteq \mathbb{R}^2$. Define an indicator variable $Z_{i,j} \in \{0,1\}$ for the distance $Y_{ij} \in \mathbb{R}^+$ between a pair of sensors (x_i, x_j) being either observed ($Z_{i,j} = 1$) or not ($Z_{i,j} = 0$), according to

$$Z_{i,j} \sim \texttt{Binom}\left(1, \exp\left(-\frac{\|x_i - x_j\|_2^2}{2R^2}\right)\right).$$

If the distance is observed, then Y_{ij} is corrupted by Gaussian noise, i.e.

$$Y_{i,j} | Z_{i,j} = 1 \sim \mathcal{N}\left(\|x_i - x_j\|, \sigma^2\right),$$

and $Y_{i,j} = 0$ otherwise.

Previously, [65] focussed on MAP estimation of the sensor locations, and [70] focussed on a well-conditioned case ($S = 8$ sensors and $B = 3$ base sensors with known locations) that results in almost no ambiguity in the posterior. We argue that Bayesian quantification of uncertainty is more important for cases where noise and missing measurements *does not* allow to reconstruct the sensor locations exactly. We therefore reuse the dataset from [70] ($R = 0.3$, $\sigma^2 = 0.02$)[1], but only use the first $S = 3$ locations/observations. In order to encourage ambiguities in the localisation task, we only use the first 2 base sensors of [70] with known locations that each observe distances to the S unknown sensors but not to each other. Unlike [70], we use a Gaussian prior $\mathcal{N}(\mathbf{0.5}, I)$ to avoid the posterior being situated in a bounded domain.

Figure 5.3 shows the marginalised posterior for one run each of KASMC (SMC) and KAMH (MCMC), with the number of iterations set to that the number of likelihood evaluations match across algorithms (500,000). We run KASMC using $10,000$ particles and a bridge length of 50, and MCMC-KAMH for $50 \times 10,000$ iterations of which we discard half as burn-in; both are initialized with samples from the prior. Tuning parameters ν^2 are set using a diminishing adaptation schedule $\lambda_t = 1/\sqrt{t}$ for KAMH and a fixed learning rate $\lambda_t = 1$ for KASMC. MCMC is not able to traverse between the multiple modes and interpretations of the data, in contrast to SMC.

A quantitative comparison of the samples is challenging due to the lack of a set of benchmark samples. Unlike for the uni-modal posterior induced by the UCI glass example from Chapter 3, merging the output of multiple MCMC chains that converged on different modes does not lead to benchmark samples from the correct distribution. Several work exists to attack such issues [70, 87]; we avoid going into details here.

[1] Downloaded from `http://www.ics.uci.edu/~slan/lanzi/CODES_files/WHMC-code.zip` on 8/Oct/2015.

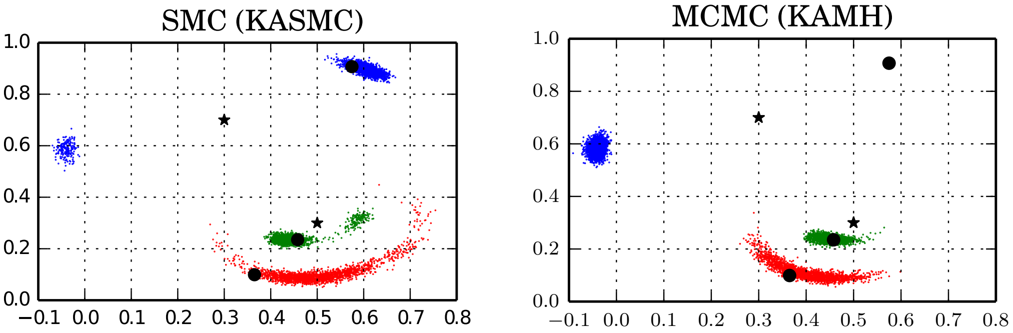


Figure 5.3: Posterior samples of unknown sensor locations (in color) by kernel-based SMC and MCMC on the sensors dataset. The set-up of the true sensor locations (black dots) and base sensors (black stars) causes uncertainty in the posterior. SMC recovers all modes while MCMC does not. The posterior has a clear non-linear structure.

5.3.3 KASMC: evidence estimation for intractable GPs

We return to the Bayesian classification problem outlined in Section 3.3.1, which has a non-linear marginal posterior over GP hyper-parameters, c.f. Figure 3.4, Figure 4.5. Where the previous experiments on that example were in the context of pseudo-marginal MCMC, the present experiment is based around nested importance sampling, c.f. Section 5.1.3 and Tran et al. [125]. Compared to previous experiments, we now emphasise a different point: KSMC's ability to estimate the model evidence as compared to KAMH. We also illustrate convergence benefits compared to ASMC.

We begin by establishing ground truth model evidence via running 20 SMC instances using $n = 1000$ particles and a bridge length of 30, and averaging their evidence estimates. We then compare the evidence estimates of all all algorithms against that reference. The experiment is performed 50 times, using $n = 100$ particles and a bridge length of 20, starting from the prior on the log hyper-parameters $\pi_0(\cdot) = \mathcal{N}(\cdot \mid 0, 5^2)$. The learning rate is constant $\lambda_t = 1$, and adaptation is towards an acceptance rate of 0.23.

Figure 5.4 shows that evidence estimates of KASMC exhibit less variance than those of ASMC, at similar levels of estimation error (not shown here).

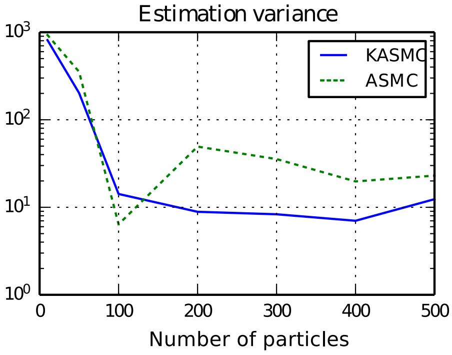


Figure 5.4: Estimating model evidence of a GP using the IS^2 framework. The plot shows the MC variance over 50 runs as a function of the size of the particle system.

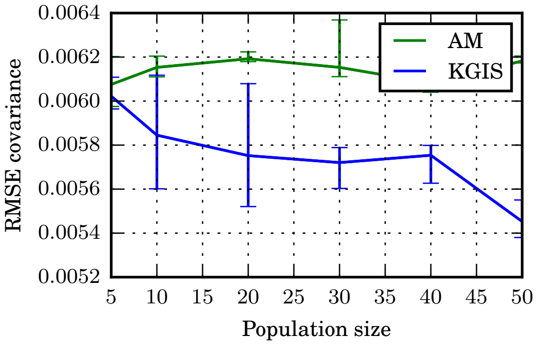


Figure 5.5: Convergence of RMSE for estimating all elements of the posterior covariance matrix of the stochastic volatility model.

5.3.4 KGRIS: evidence estimation for intractable stochastic volatility models

A particularly challenging class of Bayesian inverse problems are stochastic volatility models. As time series models, they often involve high-dimensional nuisance variables, which usually cannot be integrated out analytically. Furthermore, risk management necessitates to account for parameter and model uncertainty, and models have to capture the non-linearities in the data [33]. We here concentrate on the prediction of daily volatility of asset prices, reusing the model and dataset studied by Chopin et al. [33] to evaluate KGRIS.

The problem setting is similar to the Gaussian Process experiment in Section 3.3.1, where latent function responses of the GP are high-dimensional and cannot be be integrated out analytically. As in Section 3.3.1, we instead work with (unbiased) random estimates of the marginalised likelihood, obtained via iterated importance sampling, which results in unavailable gradients. We compare two gradient-free PMC versions: KGRIS and a random walk PMC with global covariance adaptation in the style of Haario et al. [61].

Model details

Let s_t be the value of a financial asset on day t, then $y_t = 10^{(5/2)} \log(s_t / s_{t-1})$ is called the log-returns (up-scaling for numerical reasons). We model the observed log-return y_t as dependent on a latent v_t by the observation equation

$$y_t = \mu + \beta v_t + \sqrt{v_t} \epsilon_t$$

for $t \leq 1$. Here ϵ_t is a sequence of i.i.d. standard Gaussian errors and v_t is assumed to be a stationary stochastic process known as the *actual volatility*. Chopin et al. [33] develop a hierarchical model for v_t based on the idea of analytically integrating a continuous time volatility model over daily intervals, c.f. [33] for details. Using this construction, the (discrete time) v_t is parametrised by stationary mean ξ and variance ω^2 of the so called *spot volatility* and the exponential decay λ of its auto-correlation. This results in the following model for the actual volatility v_t:

$$\begin{gathered} k \sim \texttt{Poisson}(\lambda \xi^2 / \omega^2) \\ c_1, \ldots, c_k \sim \texttt{Uniform}(t, t+1), \quad e_1, \ldots, c_k \sim \texttt{Exponential}(\xi / \omega^2) \\ z_{t+1} = z_t \exp(-\lambda) + \sum_{j=1}^{k} e_j \exp(-\lambda(t+1-c_j)) \\ v_{t+1} = \lambda^{-1} \left(z_t - z_{t+1} + \sum_{j=1}^{k} e_j \right) \end{gathered}$$

where z_t is the discretely sampled spot volatility process and $(v_{t+1}, z_{t+1})^\top$ is the Markovian representation of the state process. The variables k, $c_1, \ldots, c_k$ and $e_1, \ldots, e_k$ are generated independently for each time period t. The dynamics imply $\Gamma(\xi^2/\omega^2, \xi^2/\omega^2)$ to be the stationary distribution for z_t, which is also used as the initial distribution on z_0.

The parameters of the model are $\theta = (\mu, \beta, \xi, \omega^2, \lambda)$ and we wish to sample from their (marginalised) posterior. Despite the underlying time series, this posterior is static, c.f. Section 5.1.1. Chopin et al. [33] developed a sampler based on iterated batch importance sampling using nested SMC with pseudo-marginal MCMC moves and called their approach *SMC*^{2}. In practice, this means that each target evaluation in Algorithm 2, i.e. MCMC rejuvenation and weight computation, is based on random importance sampling estimates – whereas pseudo-marginal MCMC is to run MCMC on intractable targets, SMC^2 is running SMC on intractable targets. See [33] for more details.

Experimental details

In our experiment, we use KGRIS proposals in a population Monte Carlo setting, i.e. without resorting to MCMC moves at all. We re-use the code developed for the original SMC^2 paper to obtain likelihood estimates. The observed s_t are the 753 observations from consecutive days of the S&P 500 index also used by Chopin et al. [33]. KGRIS uses a particle system of increasing sizes with each particle going through 100 iterations. Figure 5.6 shows a plot of the pair-wise marginals of this posterior.

We use the same vague priors as Chopin et al. [33],

$$\mu \sim \mathcal{N}(0, \sigma^2 = 2), \beta \sim \mathcal{N}(0, \sigma^2 = 2), \xi \sim \text{Exp}(0.2)$$
$$\omega^2 \sim \text{Exp}(0.2), \lambda \sim \text{Exp}(1).$$

Figure 5.5 shows that the incorporated gradients lead to better performance of KGRIS in estimating the target covariance matrix. This is in-line with the finding that GRIS improves over pure random walk methods [102].

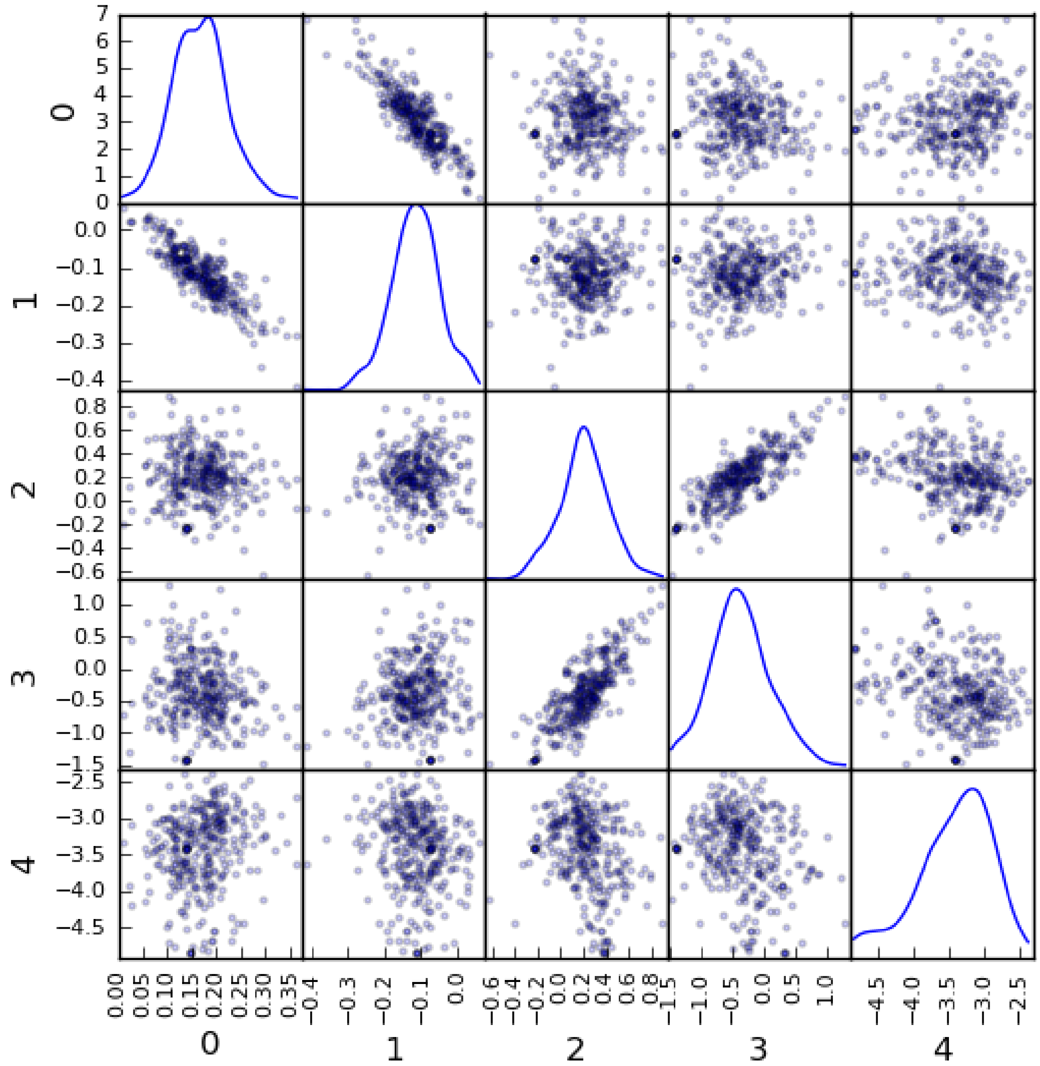


Figure 5.6: Samples from the doubly stochastic volatility model used in Section 5.3.4.

Part II

Efficient and principled score estimation

Chapter 6

Efficient and principled score estimation

This chapter is based on collaborative work, D. Sutherland, H. Strathmann, M. Arbel, and A. Gretton. "Efficient and principled score estimation". In: *arXiv preprint arXiv:1705.08360* (2017). Joint first two authors. Submitted.

We propose a fast method with statistical guarantees for learning an exponential family density model in a reproducing kernel Hilbert space. The model is learned by fitting the derivative of the log density, the *score*, thus avoiding the need to compute a normalisation constant. We improved the computational efficiency of an earlier solution with a low-rank, Nyström-like solution. The new solution retains the consistency and convergence rates of the full-rank solution, with guarantees on the degree of cost and storage reduction. We evaluate the method in experiments on density estimation and in the construction of an adaptive Hamiltonian Monte Carlo sampler. Compared to an existing score learning approach using a de-noising auto-encoder, our estimator is empirically more data-efficient when estimating the score, runs faster, and has fewer parameters which can be tuned in a principled and interpretable way.

Chapter outline

We reviewed the original estimator [117] and described several related approaches in Section 2.2. In Section 6.1, we introduce our approximation scheme and establish an estimator for it, which we analyse regarding consistency and generalisation error bounds in Section 6.2. In our experiments in Section 6.3, we compare our approach against the full solution of Sriperumbudur et al. [117], the previous heuristics score matching 'lite' and 'finite' from Chapter 4, Proposition 3 and Proposition 4 respectively, and an auto-encoder-based score estimator of Alain and Bengio [2], which we will introduce in Chapter 6.

The kernel exponential family model was introduced in Section 2.1.

Related work on fast approximate kernel regression

The system of (2.13) is related to the problem of kernel ridge regression, which suffers from similar $\mathcal{O}(n^3)$ computational cost. Thus we will briefly review methods for speeding up kernel regression.

Nyström methods

We refer here to a class of broadly related Nyström-type methods [99, 112, 130]. The representer theorem [100] guarantees that the minimiser of the empirical regression loss for a training set $X = \{X_b\}_{b \in [n]}$ over the RKHS $\mathcal{H}$ with kernel k will lie in the subspace $\mathcal{H}_X = \operatorname{span}\{k(X_b, \cdot)\}_{b \in [n]}$. Nyström methods find an approximate solution by optimizing over a smaller subspace $\mathcal{H}_Y$, usually given by $\mathcal{H}_Y = \operatorname{span}\{k(y, \cdot)\}_{y \in Y}$ for a set of m points $Y \subseteq X$ chosen uniformly at random. This decreases the computational burden both of training ($\mathcal{O}(n^3)$ to $\mathcal{O}(nm^2)$ time, $\mathcal{O}(n^2)$ to $\mathcal{O}(nm)$ memory) and testing ($\mathcal{O}(n)$ to $\mathcal{O}(m)$ time and memory).

Random feature approximations

Another popular method for scaling up kernel methods is to use random Fourier features [89, 116, 123] and their variants. Rather than finding the best solution in a subspace of $\mathcal{H}$, these methods choose a data-independent set of parametric features such that expected inner products between the features coincide with the kernel. These methods have some attractive computational properties, as for example outlined in Section 4.6.3, but generally also require the number of features to increase with the data size in a way that can be difficult to analyse: see Rudi et al. [98] for such an analysis in regression.

Sketching

Another scheme for improving the speed of kernel ridge regression, sketching [133, 134] compresses the kernel matrix and the labels by multiplying with a sketching matrix. These methods have some overlap with Nyström-type approaches, and our method will encompass certain classes of sketches [99, Appendix C.1].

Deep learning for direct score estimation

Alain and Bengio [2] proposed a deep learning-based approach to directly learn a score function from samples. De-noising auto-encoders are networks trained to recover the original inputs from versions with noise added. A de-noising auto-encoder trained with L_2 loss and noise $\mathcal{N}(0, \sigma^2 I)$ can be used to construct a score estimator:

$$(r_\sigma(x) - x)/\sigma^2 \approx \nabla_x \log p_0(x),$$

where r_σ is the auto-encoder's reconstruction function. When the auto-encoder has infinite capacity and is trained to its global optimum, Alain and Bengio [2] show that this estimator is consistent as $\sigma \to 0$. For realistic auto-encoders with finite representation capacity, however, the consistency of this approach remains an open question. Moreover, this technique has many hyper-parameters to choose, both in the architecture of the network and in how it is trained, with no theory yet available to guide those choices.

Computing un-normalised densities from a non-parametric learned score function, as the auto-encoder, can be a more challenging task. A direct approach would involve numerical integration of the score estimate, where errors can accumulate; moreover, as discussed by Alain and Bengio [2, Section 3.6], a given score estimate might not correspond to a valid gradient function, or might not yield a normalizable density.

6.1 Nyström kernel exponential families

To alleviate the computational costs of the linear system in (2.13), we apply the Nyström idea to the estimator of the full kernel exponential family model in (2.11). More precisely, we select a set of m 'basis' points $Y = \{Y_a\}_{a \in [m]}$, and restrict the optimisation in (2.11) to

$$\mathcal{H}_Y := \operatorname{span}\{\partial_i k(Y_a, \cdot)\}_{a \in [m]}^{i \in [d]}, \tag{6.1}$$

which is a subspace of $\mathcal{H}$ with elements that can be represented using md coefficients, similar to (2.11). Typically $Y \subset X$; in particular, Y is usually chosen as a uniformly random subset of X. We could, however, use any set of points Y, or even a different set of spanning vectors than $\partial_i k(Y_a, \cdot)$.

Theorem 2. *The regularised minimiser of the empirical Fisher divergence (2.10) over $\mathcal{H}_Y$ (6.1) is*

$$\begin{aligned} f_{\lambda,n}^m &= \operatorname*{argmin}_{f \in \mathcal{H}_Y} \hat{J}(f) + \frac{1}{2}\lambda \|f\|_{\mathcal{H}}^2 = \sum_{a=1}^{m} \sum_{i=1}^{d} (\beta_Y)_{(a,i)} \partial_i k(Y_b, \cdot), \\ \beta_Y &= -(\tfrac{1}{n} B_{XY}^{\mathsf{T}} B_{XY} + \lambda G_{YY})^{\dagger} h_Y. \end{aligned} \tag{6.2}$$

Here † denotes the pseudo-inverse, and $B_{XY} \in \mathbb{R}^{nd \times md}, G_{YY} \in \mathbb{R}^{md \times md}$, $h_Y \in \mathbb{R}^{md}$

are given by

$$(B_{XY})_{(b,i),(a,j)} = \partial_i \partial_{j+d} k(X_b, Y_a) \qquad (G_{YY})_{(a,i),(a',j)} = \partial_i \partial_{j+d} k(Y_a, Y_{a'})$$
$$(h_Y)_{(a,i)} = \frac{1}{n} \sum_{b=1}^{n} \sum_{j=1}^{d} \partial_i \partial_{j+d}^2 k(Y_a, X_b) + \partial_i \partial_{j+d} k(Y_a, X_b) \partial_j \log q_0(X_b).$$

The proof, which is similar to the kernel ridge regression analogue [99], is given in Chapter 7. In fact, we show a slight generalisation (Lemma 4 in Chapter 7), which also applies to more general subspaces $\mathcal{H}_Y$.

It is worth emphasizing that in order to evaluate an estimate $f_{\lambda,n}^m$, we need only evaluate derivatives of the kernel between the basis points Y and the test point x. If Y and X do not overlap, we no longer need X at all: its full contribution is summarised in β_Y. When $Y \subseteq X$, the above quantities are simply block sub-sampled versions of the terms in the full solution (2.13). Note, however, that when $Y = X$ we do not exactly recover the solution (2.13), because $\hat{\xi}$ contains components of the form $\partial_i^2 k(X_b, \cdot) \notin \mathcal{H}_X$.

Computing the the $md \times md$ matrix in (6.2) takes $\mathcal{O}(nmd^2)$ memory and $\mathcal{O}(nm^2d^3)$ time, both linear in n. Computing the pseudo-inverse takes $\mathcal{O}(m^3d^3)$ computation, independent of n. Moreover, evaluating $f_{\lambda,n}^m$ takes $\mathcal{O}(md)$ time, as opposed to the $\mathcal{O}(nd)$ time for $f_{\lambda,n}$.

6.1.1 Relationship to finite and lite kernel exponential families

In Chapter 4, we proposed two alternative heuristic approximations to the full model of (2.6), used for efficient score learning in adaptive HMC. These approaches currently lack convergence guarantees.

The *finite* form in Proposition 4 uses an m-dimensional $\mathcal{H}$, defined e.g. by random Fourier features [89], where (2.11) can be computed directly in $\mathcal{H}$ in time linear in n. We have seen that such parametric features limit the expressiveness of the model: the score estimate oscillates in regions where little or no data has been observed. In the context of Hamiltonian Monte

Carlo, this leads to poor acceptance rates when the sampler enters those regions, c.f. Section 4.3. Therefore, we do not further pursue this approach in the present work here.

The *lite* approximation in Proposition 3 instead finds an estimator $f \in \operatorname{span}\{k(x,\cdot)\}_{x\in X}$. This has a similar spirit to Nyström approaches, but with a different basis from (2.11), which is based on kernel *derivatives*. Furthermore, the lite approximation uses the entirety of X, so the dependence on n is improved only by simple sub-sampling. Finally, the estimator in Proposition 3 is solely for the specific case of Gaussian kernels. The generalised version of Theorem 2 (Lemma 4 in Section 7.2) covers the basis used by the lite approximation, allowing us to generalise this method to basis sets $Y \neq X$ and to kernels other than the Gaussian. We discuss this in more detail in Part IV.

6.2 Theory

We analyse the performance of our estimator in the well-specified case: assuming that the true density p_0 is in $\mathcal{P}$ (and thus corresponds to some $f_0 \in \mathcal{H}$), we obtain both the parameter convergence of $f^m_{\lambda,n}$ to f_0 and the convergence of the corresponding density $p_{f^m_{\lambda,n}}$ to the true density p_0. Detailed proofs are provided in Chapter 7.

Theorem 3. *Assume the conditions listed in Section 7.1.1 (similar to those of Sriperumbudur et al. [117] for the well-specified case), and use the $\mathcal{H}_Y$ of* (6.1) *with the basis set Y chosen uniformly at random from the size-m subsets of the training set X. Let $\beta \geq 0$ be the range-space smoothness parameter of the true density f_0, and define $b = \min\left(\beta, \frac{1}{2}\right)$, $\theta = \frac{1}{2(b+1)} \in [\frac{1}{3}, \frac{1}{2}]$. As long as $m = \Omega\left(n^{\theta}\log n\right)$, then with $\lambda = n^{-\theta}$ we obtain*

$$\|f^m_{\lambda,n} - f_0\|_{\mathcal{H}} = \mathcal{O}_{p_0}\left(n^{-\frac{b}{2(b+1)}}\right), \qquad J(p_0\|p_{f^m_{\lambda,n}}) = \mathcal{O}_{p_0}\left(n^{-\frac{2b+1}{2(b+1)}}\right).$$

The first statement implies that $p_{f^m_{\lambda,n}}$ also converges to p_0 in L_r $(1 \leq r \leq \infty)$ and

Hellinger distances at a rate $\mathcal{O}_{p_0}\left(n^{-\frac{b}{2(b+1)}}\right)$*, and that* $\mathrm{KL}(p_0 \| p_{f^m_{\lambda,n}}), \mathrm{KL}(p_{f^m_{\lambda,n}} \| p_0)$ *are each* $\mathcal{O}_{p_0}\left(n^{-\frac{b}{b+1}}\right)$.

The rate of convergence in J exactly matches the rate for the full-data estimator $f_{\lambda,n}$ shown by [117] in J; the rates in other divergences essentially match, except that ours saturate slightly sooner as β increases. Thus, for any problem satisfying the assumptions, we can achieve the same statistical properties as the full-data setting with $m = \Omega\left(\sqrt{n}\log n\right)$, while in the smoothest problems we need only $m = \Omega\left(n^{1/3}\log n\right)$. A finite-sample version of Theorem 3, with explicit constants, is stated and proved in Section 7.3.

6.2.1 Reduced costs compared to kernel ridge regression

This substantial reduction in computational expense is in contrast to the comparable analysis for kernel ridge regression [99], which for the hardest problems requires $m = \Omega(n \log n)$, giving no computational savings at all. In the best general case, it also needs $m = \Omega(n^{1/3} \log n)$. This rate was itself a significant advance: a prior analysis based on stability of the kernel approximation [37] results in a severe additional penalty when using Nyström, matching the worst-case error rates for the full solution, yet still requiring $m = \Omega(n)$ (i.e., according to the earlier reasoning, we would not be guaranteed to benefit from improved rates in easier problems).

6.2.2 Proof outline

Our proof uses techniques both from the analysis of the full-data estimator [117] and from an analysis of generalisation error for Nyström-subsampled kernel ridge regression [99].

Each of the losses considered in Theorem 3 can be bounded in terms of $\|f - f_0\|_{\mathcal{H}}$. We decompose this loss relative to $f^m_\lambda = \operatorname{argmin}_{f \in \mathcal{H}_Y} J(f) + \frac{1}{2}\lambda \|f\|^2_{\mathcal{H}}$, which is the best regularised estimator in population with the particular basis Y. The decomposition is,

$$\|f^m_{\lambda,n} - f_0\|_{\mathcal{H}} \leq \|f^m_{\lambda,n} - f^m_\lambda\|_{\mathcal{H}} + \|f^m_\lambda - f_0\|_{\mathcal{H}}. \tag{6.3}$$

The first term on the right-hand side of (6.3) is the *estimation error*, which represents our error due to having a finite number of samples n: this term decreases as $n \to \infty$, but it will increase as $\lambda \to 0$. It could conceivably increase as $m \to \infty$ as well, but we show using concentration inequalities in $\mathcal{H}$ that no matter the m, the estimation error is $\mathcal{O}_{p_0}\left(\frac{1}{\lambda\sqrt{n}}\right)$.

The last term of (6.3) is the *approximation error*, where 'approximation' refers both to the regularisation by λ and the restriction to the subspace $\mathcal{H}_Y$. This term is independent of n; it decreases as $\mathcal{H}_Y$ grows (i.e. as $m \to \infty$), and also decreases as $\lambda \to 0$, as we allow ourselves to more directly minimise the population risk. The key to bounding this term is to exploit the nature of the space $\mathcal{H}_Y$. This can be done by analogy with the treatment of the 'computational error' term of Rudi et al. [99], where we show that any components of f_0 not lying within $\mathcal{H}_Y$ are relatively small in the parts of the space we observe; this is the only step of the proof that depends on the specific basis $\mathcal{H}_Y$. Having handled this contribution, we show that the approximation error term is $\mathcal{O}_{p_0}\left(\lambda^b\right)$ as long as $m = \Omega\left(\frac{1}{\lambda}\log\frac{1}{\lambda}\right)$.

The decay of the two terms is then optimised when $\lambda = n^\theta$, with θ as given in the proof.

The rate in the Fisher divergence J is better because that metric is weighted towards points in the space where we actually see data, as opposed to uniformly across the space as in (6.3). Our proof technique, similarly to that of Sriperumbudur et al. [117], allows us to account for this with an improved dependence on λ in the evaluation of both estimation and approximation errors.

6.2.3 The missing ξ

We previously noted that using $Y = X$ does not yield an identical estimator, $f^n_{\lambda,n} \neq f_{\lambda,n}$. In fact, we could achieve this by additionally including $\hat{\xi}$ within the space (6.1); it would also not be too hard to alter our proof to account for this, achieving the same asymptotic rates. Since evaluating $\hat{\xi}$ requires touching all the data points, however, we would lose the test-time improvements in both computation and memory achieved by the estimator of Theorem 2. Moreover, the experiments of Section 6.3 show that dropping $\hat{\xi}$ from the basis does not seem to be harmful in practice.

6.3 Experiments

We now validate our estimator empirically, considering two problem settings: score function estimation for known, multi-modal densities in high dimensions, and again Hamiltonian Monte Carlo, where the score is used in proposing Metropolis-Hastings moves. We first consider synthetic densities in Section 6.3.1, where we know the true densities and can evaluate convergence of the score estimates analytically with (2.8). In Section 6.3.2 we evaluate our estimator in the gradient-free Hamiltonian Monte Carlo setting from Chapter 4, where (in the absence of a ground truth) we compare the efficiency of the resulting sampler.

For all exponential family variants, we take q_0 to be a uniform distribution with support encompassing the samples, and use a Gaussian kernel $k(x,y) = \exp\left(-\|x-y\|^2/\sigma\right)$, with a tuned bandwidth σ and regularisation parameter λ, c.f. Section 4.3. We compare the following models:

(i) The **full** model, (2.11) and (2.13), from [117].

(ii) The **lite** model from Proposition 3 in Chapter 4, which sub-samples the dataset X to size m, and uses the basis $\{k(X_a, \cdot)\}$, ignoring the remaining data points (unlike the Nyström case). The code uses the regularisation $\lambda(\|f\|^2_{\mathcal{H}} + \|\beta\|^2_2)$

(iii) The **nyström** estimator of Theorem 2, choosing m distinct data points uniformly at random for Y. For numerical stability, we add $10^{-5}I$ to the matrix being inverted in (6.2), corresponding to a small L_2 regulariser on the weights β.

(iv) The **dae** model of Alain and Bengio [2], where we train a two-layer de-noising auto-encoder, with tanh code activations and linear decoding. We train with decreasing noise levels (100σ, 10σ, σ), using up to 1000 iterations of BFGS each. We tune the number of hidden units and σ, since, while [2] recommend simply choosing some small σ, this plays a similar role to a RBF kernel bandwidth, and its careful choice is essential. We differentiate (using the automatic differentiation capabilities of TensorFlow [1]) the score estimate to obtain the second derivative needed to evaluate (2.9).

6.3.1 Score convergence on synthetic densities

We first consider two synthetic densities, whose true score functions are available: The 'ring' dataset, Figure 6.1 (left), takes inspiration from the 'spiral' dataset of Alain and Bengio [2, Figure 5], being a similarly-shaped distribution but possessing a probability density for evaluation purposes. We sample points along three circles in $\mathbb{R}^2$ with radii $(1,3,5)$, with mixture weights proportional to the circumference of each circle. We then add Gaussian noise of standard deviation 0.1 in the radial direction. Extra dimensions consist of independent Gaussian noise with standard deviation 0.1. The 'grid' dataset, Figure 6.1 (right), generalises the 2-component mixture example of Sriperumbudur et al. [117, Figure 1] into a more challenging d-dimensional mixture. We first pick d random (using a fixed seed) vertices of a d-dimensional hypercube, and then construct a mixture of normal distributions, one at each selected vertex. For each run, we generate $n = 500$ training points and estimate the score on 1500 (grid) or 5000 (ring) newly generated test points. In both cases, we estimate the true score

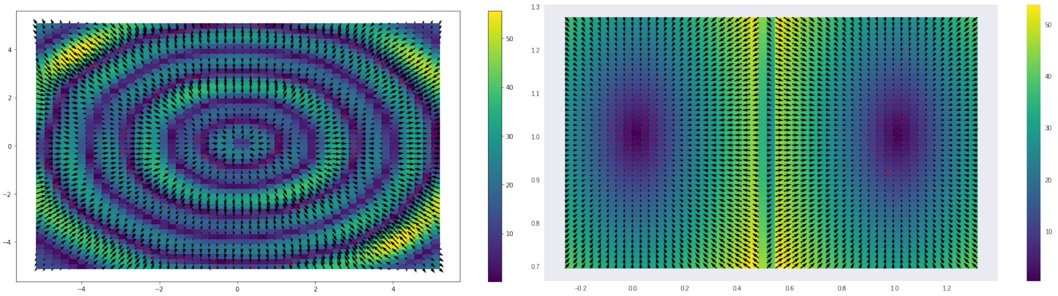


Figure 6.1: Two dimensional versions of the ring and grid dataset. The heatmap corresponds to the log-density, arrows represent the (re-scaled) score gradient field, and the red dots are samples from the density.

(2.8) on these test points to ensure a 'best case' comparison of the models, though we confirmed that using (2.9) leads to indistinguishable parameter selections and performance. For **lite** and **nyström**, we independently tune the parameters for each sub-sampling level. We report performances for the best parameters found for each method.

Results

Figure 6.2 shows convergence of the score as the dimension increases. On both the ring and grid datasets, **nyström** performs very close to the full solution, in addition to large computational savings. With reasonable performance penalties at $m = 42$, we achieve a major reduction in cost and storage over the original $n = 500$ sample size. The **lite** performance is similar to that of **nyström** at comparable levels of data retention. As expected, the performance of **nyström** gets closer to that of **full** as m increases towards n. The auto-encoder performs consistently worse than any of the kernel models, on both datasets. Auto-encoder results are also strongly clustered, with only small performance improvements as the number of hidden units increases. For the grid data, we observe that in 20 dimensions, all solutions start to converge to a similar score. This indicates that none of the methods are able to learn the structure for this number of training points and dimensions, and all solutions effectively revert to smooth, uninformative estimates.

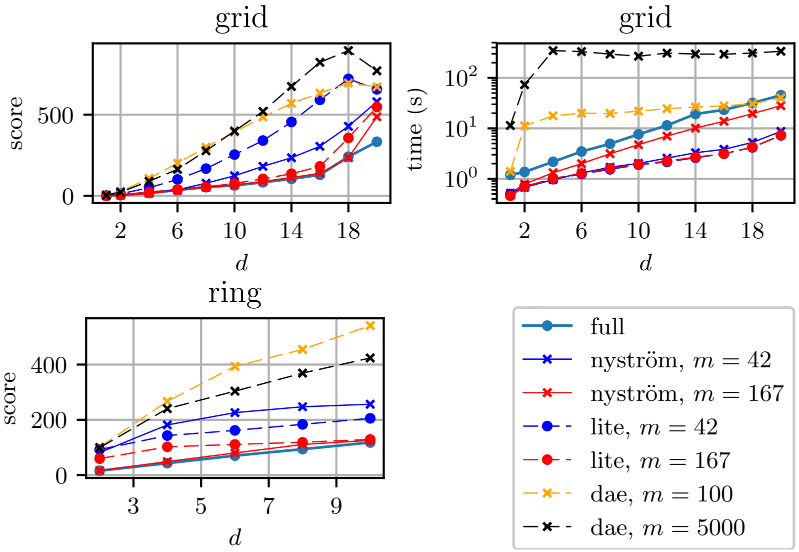


Figure 6.2: Convergence and timing on synthetic ring and grid data.

Computational costs

In respect of computational cost[1], the **lite** solution does best, followed by **nyström** for low to moderate m, with significant savings over the full solution even at $m = 167$ on the grid, and across all m on the ring. The additional cost of **nyström** over **lite** arises since it must compute all derivatives at the retained samples. The auto-encoder runtime is longer than the other methods, although we point out that the settings recommended in Alain and Bengio [2] are not optimised for run-time. We observed, however, that replacing BFGS with stochastic gradient descent or avoiding the 'decreasing noise' schedule both lead to instabilities in the solution.

6.3.2 Gradient-free Hamiltonian Monte Carlo

Our final experiment uses the methodology of Chapter 4 for constructing a gradient-free HMC sampler using score estimates learned on the previous MCMC samples. Our goal is to efficiently sample from the marginal posterior over hyper-parameters of a Gaussian process (GP) classifier on the UCI Glass dataset [76], c.f. Section 3.3.1. Once again, closed-form expressions for the score (and therefore HMC itself) are not available, due to the in-

[1] All experiments are conducted in a single CPU thread for timing comparisons, although multi-core parallelism is straightforward for all models.

tractability of the marginal data likelihood given the hyper-parameters. We compare all score estimators' ability to generate an HMC-like proposal as described in Chapter 4. An accurate score estimate would result in proposals close to an idealised HMC move, which would have a high acceptance probability. Thus, higher acceptance rates indicate better score estimates.

Kernel induced Hamiltonian flow

Our experiment assumes the idealised scenario where a burn-in is successfully completed, just like the trajectory experiments in Section 4.4.1. We run 40 random walk adaptive-Metropolis MCMC samplers for 30 000 iterations, discard the first 10 000 samples, and thin by a factor of 400. Merging these samples results in 2 000 posterior samples. We fit all score estimators on a random subset of $n = 500$ of these samples, and use the remaining 1500 samples to tune the model hyper-parameters. The validation surface obtained for **nyström** by the estimated score objective on the held-out set is shown in Figure 6.3: we note that it is smooth and easily optimised. For the **dae** (validation surface not shown here), a well-tuned level of corruption noise is essential. Starting from a random point of the initial posterior sketch, we construct trajectories along the kernel induced Hamiltonian flow, Section 4.1, using 100 steps of size 0.1, and a standard Gaussian momentum. We compute the hypothetical acceptance probability (4.2) for each step, and average over the trajectory.

Results

Figure 6.3 shows the results averaged over 200 repetitions. As before, **nyström** matches the performance of **full** for $m = n = 500$, while for $m = 100$ it attains a high acceptance rate at a considerably reduced computational cost. It also reliably outperforms **lite** for lower m, which might occur since **lite** sub-samples the data while **nyström** only sub-samples the basis. The **dae** does relatively poorly, despite a large grid-search for its hyper-parameters. For any of the models, un-tuned hyper-parameters easily lead to an acceptance rate close to zero.

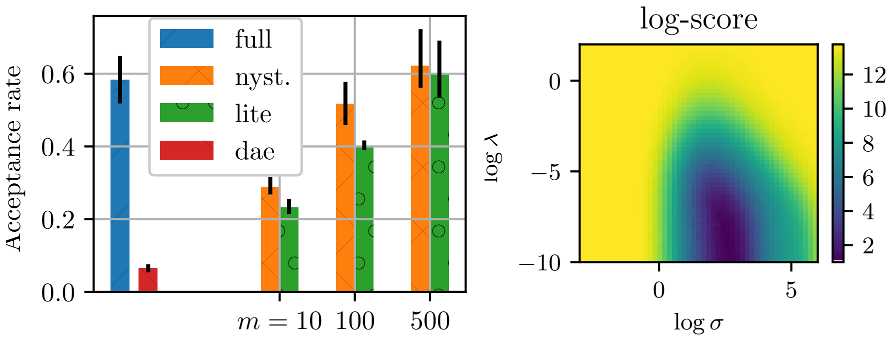


Figure 6.3: Left: HMC acceptance rate on the GP-glass posterior, with 90% quantiles. Right: Hyper-parameter surface of the score for **nyström** with $m = 42$.

Chapter 7

Proofs

We now prove Theorems 2 and 3 from Chapter 6, as well as providing a finite-sample bound with explicit constants (Theorem 4). The chapter is based on the same collaborative work as Chapter 6.

Overview

In Section 7.1, we begin with a review of necessary notation and definitions of all necessary objects, and a review of theory for the full kernel exponential family estimator by Sriperumbudur et al. [117]. Some of Section 7.1 in redundant with Sections 2.1 and 2.2, and Chapter 6, but we collect everything here to improve readability. In Section 7.2, we establish a representer theorem for our Nyström estimator and prove Theorem 2. We address consistency and convergence in Section 7.3, by first decomposing and bounding the error in Section 7.3.1, then developing probabilistic inequalities in Section 7.3.2, and finally collecting everything into a final bound to prove Theorem 3 in Section 7.3.3. Section 7.4 establishes auxiliary results used in the proofs, in particular a concentration inequality for sums of correlated random operators in Section 7.4.2.

7.1 Preliminaries

We first establish some definitions that are useful throughout, as well as reviewing relevant results by Sriperumbudur et al. [117].

Notation

Following Section 2.1, denote by $\mathcal{H}$ a reproducing kernel Hilbert space of functions $\Omega \subseteq \mathbb{R}^d \to \mathbb{R}$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and norm $\|\cdot\|_{\mathcal{H}}$, with a kernel $k : \Omega \times \Omega \to \mathbb{R}$ given by the reproducing property (2.1), with derivatives (2.3).

We use $\|\cdot\|$ to denote the operator norm

$$\|A\| = \sup_{f : \|f\|_{\mathcal{H}} \leq 1} |\langle f, Af \rangle_{\mathcal{H}}|,$$

and A^* for the adjoint of an operator $A : \mathcal{H}_1 \to \mathcal{H}_2$,

$$\langle Af, g \rangle_{\mathcal{H}_2} = \langle f, A^* g \rangle_{\mathcal{H}_1}.$$

$\lambda_{\max}(A)$ denotes the algebraically largest eigenvalue of A. For elements $f \in \mathcal{H}_1$, $g \in \mathcal{H}_2$ recall the tensor product $f \otimes g$ (2.5), viewed as an operator from $\mathcal{H}_2$ to $\mathcal{H}_1$, where we have that $(f \otimes g)^* = g \otimes f$ and that $A(f \otimes g)B = (Af) \otimes (B^* g)$.

$C^1(\Omega)$ denotes the space of continuously differentiable functions on Ω, and $L^r(\Omega)$ the space of r-power Lebesgue-integrable functions.

As in Chapter 6, $x_{(a,i)}$ denotes $x_{(a-1)d+i}$.

Definitions

We now define a number of operators and other objects used in our study, some of which were already introduced in Section 2.2.

Definition 1. *Suppose we have a sample set $X = \{X_a\}_{a \in [n]} \subset \mathbb{R}^d$. For any $\lambda > 0$,* *define the following:*

$$C = \mathbb{E}_{x\sim p_0}\left[\sum_{i=1}^{d} \partial_i k(x,\cdot)\otimes\partial_i k(x,\cdot)\right] : \mathcal{H}\to\mathcal{H}; \qquad C_\lambda = C+\lambda I \quad (7.1)$$

$$\xi = -Cf_0 = \mathbb{E}_{x\sim p_0}\left[\sum_{i=1}^{d} \partial_i k(x,\cdot)\partial_i \log q_0(x) + \partial_i^2 k(x,\cdot)\right] \in \mathcal{H} \quad (7.2)$$

$$Z_X = \sum_{b=1}^{n}\sum_{i=1}^{d} e_{(b,i)}\otimes\partial_i k(X_b,\cdot) : \mathcal{H}\to\mathbb{R}^{nd}; \quad (7.3)$$

here $e_{(b,i)}$ has component $(b-1)d+i$ equal to 1 and all others 0.

Define estimators of (7.1) and (7.2) by

$$\hat{C} = \frac{1}{n}Z_X^* Z_X = \frac{1}{n}\sum_{a=1}^{n}\sum_{i=1}^{d}\partial_i k(X_a,\cdot)\otimes\partial_i k(X_a,\cdot) : \mathcal{H}\to\mathcal{H} \quad (7.4)$$

$$\hat{\xi} = \frac{1}{n}\sum_{a=1}^{n}\sum_{i=1}^{d}\partial_i k(X_a,\cdot)\partial_i \log q_0(X_a) + \partial_i^2 k(X_a,\cdot) \in \mathcal{H}, \quad (7.5)$$

and $\hat{C}_\lambda := \hat{C}+\lambda I$. The operator Z_X in (7.3) evaluates derivatives, $(Z_X f)_{(b,i)} = \partial_i f(X_b)$, whereas for $\alpha\in\mathbb{R}^{nd}$, $Z_X^*\alpha = \sum_{b=1}^{n}\sum_{i=1}^{d}\alpha_{(b,i)}\partial_i k(X_b,\cdot)$. C is similar to the standard covariance operator in similar analyses, c.f. Section 2.2 and [27, 99].

Analogous to [99], we also use these C-derived quantities:

$$\mathcal{N}_\infty(\lambda) := \sup_{x\in\Omega}\sum_{i=1}^{d}\left\|C_\lambda^{-\frac{1}{2}}\partial_i k(x,\cdot)\right\|_{\mathcal{H}}^2$$

$$\mathcal{N}'_\infty(\lambda) := \sup_{\substack{x\in\Omega\\ i\in[d]}}\left\|C_\lambda^{-\frac{1}{2}}\partial_i k(x,\cdot)\right\|_{\mathcal{H}}^2 .$$

Note that under (**G**), $\mathcal{N}_\infty(\lambda)\le d\mathcal{N}'_\infty(\lambda)\le\frac{d\kappa_1^2}{\lambda}$, and $\|C\|\le d\kappa_1^2$.

7.1.1 Assumptions

We need the following assumptions on p_0, q_0, and $\mathcal{H}$. These assumptions, or closely related ones, were all used by [117] for various parts of their analysis. Assumptions (**B**) to (**D**) ensure that the form for the Fisher divergence

$J(p_0\|p)$ in (2.9) is valid. Assumption (**E**) implies $J(p_0\|p_f)$ is finite for any $p_f \in \mathcal{P}$. Assumption (**G**) is used to get probabilistic bounds on the convergence of the estimators, and implies Assumption (**E**). Note that $\kappa_2^2 < \infty$ and $Q < \infty$ can be replaced by $L^2(\Omega, p_0)$ integrability assumptions as in [117] without affecting the asymptotic rates, but $\kappa_1^2 < \infty$ is used to get Nyström-like rates. Assumption (**H**) is additionally needed for the convergence in L^r, Hellinger, and KL distances.

(**A**) (Well-specified) The true density is $p_0 = p_{f_0} \in \mathcal{P}$, for some $f_0 \in \mathcal{F}$.

(**B**) $\operatorname{supp} p_0 = \Omega$ is a non-empty open subset of $\mathbb{R}^d$, with a piecewise smooth boundary $\partial\Omega := \bar{\Omega} \setminus \Omega$, where $\bar{\Omega}$ denotes the closure of Ω.

(**C**) p_0 is continuously extensible to $\bar{\Omega}$. k is twice continuously differentiable on $\Omega \times \Omega$, with $\partial^{\alpha,\alpha}k$ continuously extensible to $\bar{\Omega} \times \bar{\Omega}$ for $|\alpha| \leq 2$.

(**D**) $\partial_i\partial_{i+d}k(x,x')|_{x'=x}p_0(x) = 0$ for $x \in \partial\Omega$, and for all sequences of $x \in \Omega$ with $\|x\|_2 \to \infty$ we have have $p_0(x)\sqrt{\partial_i\partial_{i+d}k(x,x')}\Big|_{x'=x} = o\left(\|x\|^{1-d}\right)$ for each $i \in [d]$.

(**E**) (Integrability) For all $i \in [d]$, each of

$$\partial_i\partial_{i+d}k(x,x')|_{x'=x}, \; \sqrt{\partial_i^2\partial_{i+d}^2k(x,x')}\Big|_{x'=x}, \; \partial_i \log q_0(x)\sqrt{\partial_i^2\partial_{i+d}^2k(x,x')}\Big|_{x'=x}$$

are in $L^1(\Omega, p_0)$, and $q_0 \in C^1(\Omega)$.

(**F**) (Range space) $f_0 \in \operatorname{range}(C^\beta)$ for some $\beta \geq 0$, and $\left\|C^{-\beta}f_0\right\|_{\mathcal{H}} < R$ for some $R < \infty$. The operator C was defined by (7.1).

(**G**) (Bounded derivatives) $\operatorname{supp}(q_0) = \mathcal{H}$, and the following quantities are finite:

$$\kappa_1^2 := \sup_{\substack{x\in\Omega\\ i\in[d]}} \partial_i\partial_{i+d}k(x,x')\big|_{x'=x}, \quad \kappa_2^2 := \sup_{\substack{x\in\Omega\\ i\in[d]}} \partial_i^2\partial_{i+d}^2k(x,x')\Big|_{x'=x}$$

$$Q := \sup_{\substack{x\in\Omega\\ i\in[d]}} |\partial_i \log q_0(x)|\,.$$

(H) (Bounded kernel) $\kappa^2 := \sup_{x\in\Omega} k(x,x)$ is finite.

Full-data result

We review a result for the full estimator established by Sriperumbudur et al. [117, Theorem 3].

Lemma 1. *Under Assumptions (**A**) to (**E**),*

$$J(f) = J(p_0\|p_f) = \frac{1}{2}\langle f - f_0, C(f - f_0)\rangle_{\mathcal{H}} = \frac{1}{2}\langle f, Cf\rangle_{\mathcal{H}} + \langle f, \xi\rangle_{\mathcal{H}} + J(p_0\|q_0).$$

Thus for $\lambda > 0$, the unique minimizer of the regularised loss function $J_\lambda(f) = J(f) + \frac{1}{2}\lambda\|f\|_{\mathcal{H}}^2$ is

$$f_\lambda = \operatorname*{argmin}_{f\in\mathcal{H}} J_\lambda(f) = -C_\lambda^{-1}\xi = C_\lambda^{-1}Cf_0.$$

Using the estimators (7.4) and (7.5), define an empirical estimator of the loss function (2.10), up to the additive constant $J(p_0\|q_0)$, as

$$\hat{J}(f) = \frac{1}{2}\langle f, \hat{C}f\rangle_{\mathcal{H}} + \langle f, \hat{\xi}\rangle_{\mathcal{H}}.$$

There is a unique minimizer of $\hat{J}_\lambda(f) = \hat{J}(f) + \frac{1}{2}\lambda\|f\|_{\mathcal{H}}^2$:

$$f_{\lambda,n} = \operatorname*{argmin}_{f\in\mathcal{H}} \hat{J}_\lambda(f) = -\hat{C}_\lambda^{-1}\hat{\xi}.$$

$f_{\lambda,n}$ can be computed according to [117, Theorem 4].

We next define sub-sampling versions of the operators in Definition 1. Note, however, that we consider a more general $\mathcal{H}_Y$ than (6.1), allowing to use any finite-dimensional subspace of $\mathcal{H}$.

Definition 2 (Sub-sampling operators). *Let $Y = \{y_a\}_{a\in[m]} \subset \mathcal{H}$ be some basis set, and let its span be $\mathcal{H}_Y = \operatorname{span}(Y)$; note that (6.1) uses $y_{(a,i)} = \partial_i k(Y_a, \cdot)$. Then define*

$$Z_Y = \sum_{a=1}^{m} e_a \otimes y_a : \mathcal{H} \to \mathbb{R}^m. \tag{7.6}$$

Let Z_Y have singular value decomposition $Z_Y = U\Sigma V^*$, where $\Sigma \in \mathbb{R}^{t\times t}$ for some $t \leq m$. For an operator $A : \mathcal{H} \to \mathcal{H}$, let

$$g_Y(A) = V(V^*AV)^{-1}V^*. \tag{7.7}$$

Here,

$$P_Y := VV^*$$

is the orthogonal projection operator onto $\mathcal{H}_Y$, while V^*V is the identity on $\mathbb{R}^t$. As Z_X in (7.3), the operator Z_Y in (7.6) evaluates (here sub-sampled) derivatives.

The projected inverse function g_Y, defined by Rudi et al. [99], is crucial in our study, and so we first establish useful properties of it.

Lemma 2 (Properties of g_Y). *Let $A : \mathcal{H} \to \mathcal{H}$ be a positive operator, and define $A_\lambda = A + \lambda I$ for any $\lambda > 0$. The operator g_Y of (7.7) satisfies the following:*

(i) $g_Y(A)P_Y = g_Y(A)$

(ii) $P_Y g_Y(A) = g_Y(A)$

(iii) $g_Y(A_\lambda)A_\lambda P_Y = P_Y$

(iv) $g_Y(A_\lambda) = (P_Y A P_Y + \lambda I)^{-1} P_Y$

(v) $\|A_\lambda^{\frac{1}{2}} g_Y(A_\lambda) A_\lambda^{\frac{1}{2}}\| \leq 1$

Proof. (i) and (ii) follow from $V^* P_Y = V^* V V^* = V^*$ and $P_Y V = V V^* V = V$, respectively. (iii) is similar: $g_Y(A_\lambda) A_\lambda P_Y = V(V^* A_\lambda V)^{-1} V^* A_\lambda V V^* = V V^*$. For (iv),

$$P_Y = V V^* = V(V^* A_\lambda V)(V^* A_\lambda V)^{-1} V^* = V(V^* A_\lambda V) V^* V (V^* A_\lambda V)^{-1} V^*.$$

But $V(V^* A_\lambda V) V^* = V(V^* A V + \lambda V^* V) V^* = (P_Y A P_Y + \lambda I) P_Y$, so we have

$$P_Y = (P_Y A P_Y + \lambda I) P_Y g_Y(A_\lambda);$$

left-multiplying both sides by $(P_Y A P_Y + \lambda I)^{-1}$ and using (ii) yields the desired result. Finally,

$$\begin{aligned}\left(A_\lambda^{\frac{1}{2}} g_Y(A_\lambda) A_\lambda^{\frac{1}{2}}\right)^2 &= A_\lambda^{\frac{1}{2}} g_Y(A_\lambda) A_\lambda g_Y(A_\lambda) A_\lambda^{\frac{1}{2}} \\ &= A_\lambda^{\frac{1}{2}} V(V^* A_\lambda V)^{-1} V^* A_\lambda V (V^* A_\lambda V)^{-1} V^* A_\lambda^{\frac{1}{2}} \\ &= A_\lambda^{\frac{1}{2}} V(V^* A_\lambda V)^{-1} V^* A_\lambda^{\frac{1}{2}} \\ &= A_\lambda^{\frac{1}{2}} g_Y(A_\lambda) A_\lambda^{\frac{1}{2}},\end{aligned}$$

so that $A_\lambda^{\frac{1}{2}} g_Y(A_\lambda) A_\lambda^{\frac{1}{2}}$ is a projection. Thus its operator norm is either 0 or 1, and (v) follows. □

7.2 Representer theorem for Nyström

In this section, we cover Theorem 2. We first establish some representations for $f_{\lambda,n}^m$ in terms of operators on $\mathcal{H}$ (in Lemma 3), and then show Lemma 4, which generalizes Theorem 2. This parallels Rudi et al. [99, Appendix C].

Lemma 3. *Under Assumptions (**A**) to (**E**), the unique minimizer of $\hat{J}(f) + \lambda \|f\|_{\mathcal{H}}^2$ in $\mathcal{H}_Y$ is*

$$f_{\lambda,n}^m := \operatorname*{argmin}_{f \in \mathcal{H}_Y} \hat{J}_\lambda(f) = -(P_Y \hat{C} P_Y + \lambda I)^{-1} P_Y \hat{\xi} = -g_Y(\hat{C}_\lambda) \hat{\xi}. \tag{7.8}$$

Proof. We begin by rewriting the minimisation using Lemma 1 as

$$\begin{aligned}
f^m_{\lambda,n} &= \operatorname*{argmin}_{f\in\mathcal{H}_Y} \hat{J}_\lambda(f)\\
&= \operatorname*{argmin}_{f\in\mathcal{H}_Y} \frac{1}{2}\langle f,\hat{C}f\rangle_{\mathcal{H}} + \langle f,\hat{\xi}\rangle_{\mathcal{H}} + \frac{1}{2}\lambda\|f\|^2_{\mathcal{H}}\\
&= \operatorname*{argmin}_{f\in\mathcal{H}_Y} \frac{1}{2}\langle P_Y f,\hat{C}P_Y f\rangle_{\mathcal{H}} + \langle P_Y f,\hat{\xi}\rangle_{\mathcal{H}} + \frac{1}{2}\lambda\|f\|^2_{\mathcal{H}}\\
&= \operatorname*{argmin}_{f\in\mathcal{H}_Y} \frac{1}{2}\left\langle \frac{1}{\sqrt{n}}Z_X P_Y f, \frac{1}{\sqrt{n}}Z_X P_Y f\right\rangle_{\mathcal{H}} + \langle f,P_Y\hat{\xi}\rangle_{\mathcal{H}} + \frac{1}{2}\lambda\|f\|^2_{\mathcal{H}}\\
&= \operatorname*{argmin}_{f\in\mathcal{H}_Y} \frac{1}{2}\left\|\frac{1}{\sqrt{n}}Z_X P_Y f\right\|^2_{\mathcal{H}} + \lambda\left\langle f,\frac{1}{\lambda}P_Y\hat{\xi}\right\rangle_{\mathcal{H}} + \frac{1}{2}\lambda\|f\|^2_{\mathcal{H}} + \frac{1}{2}\lambda\left\|\frac{1}{\lambda}P_Y\hat{\xi}\right\|^2_{\mathcal{H}}\\
&= \operatorname*{argmin}_{f\in\mathcal{H}_Y} \frac{1}{2}\left\|\frac{1}{\sqrt{n}}Z_X P_Y f\right\|^2_{\mathcal{H}} + \frac{1}{2}\lambda\left\|f + \frac{1}{\lambda}P_Y\hat{\xi}\right\|^2_{\mathcal{H}}.
\end{aligned}$$

This problem is strictly convex and coercive, thus a unique $f^m_{\lambda,n}$ exists. Now, for any $f\in\mathcal{H}$, we have

$$\left\|f + \frac{1}{\lambda}P_Y\hat{\xi}\right\|^2_{\mathcal{H}} = \left\|P_Y f + \frac{1}{\lambda}P_Y\hat{\xi}\right\|^2_{\mathcal{H}} + \|(I-P_Y)f\|^2_{\mathcal{H}},$$

so that the problem

$$\operatorname*{argmin}_{f\in\mathcal{H}} \frac{1}{2}\left\|\frac{1}{\sqrt{n}}Z_X P_Y f\right\|^2_{\mathcal{H}} + \frac{1}{2}\lambda\left\|f + \frac{1}{\lambda}P_Y\hat{\xi}\right\|^2_{\mathcal{H}}$$

will yield a solution in $\mathcal{H}_Y$. This problem is also strictly convex and coercive, so its unique solution must be $f^m_{\lambda,n}$. By differentiating the objective, we can then see that

$$\begin{gathered}
\tfrac{1}{n}P_Y Z^*_X Z_X f^m_{\lambda,n} + \lambda f^m_{\lambda,n} + P_Y\hat{\xi} = 0\\
\left(P_Y\hat{C}P_Y + \lambda I\right) f^m_{\lambda,n} = -P_Y\hat{\xi},
\end{gathered}$$

which since $\hat{C}$ is positive yields the first equality of (7.8). The second follows from Lemma 2 (iv). □

Lemma 4 (Generalisation of Theorem 2)**.** *Under Assumptions (**A**) to (**E**), $f^m_{\lambda,n}$ can be computed as*

$$\begin{aligned} f^m_{\lambda,n} &= Z_Y^* \beta_Y = \sum_{a=1}^m (\beta_Y)_a y_a \\ \beta_Y &= -(\tfrac{1}{n} B_{XY}^\top B_{XY} + \lambda G_{YY})^\dagger h_Y, \end{aligned} \tag{7.9}$$

where $B_{XY} \in \mathbb{R}^{nd \times m}, G_{YY} \in \mathbb{R}^{m \times m}$, $h_Y \in \mathbb{R}^m$ are given by

$$\begin{aligned} (B_{XY})_{(b,i),a} &= \langle \partial_i k(X_b, \cdot), y_a \rangle_{\mathcal{H}} \qquad (7.10) \\ (G_{YY})_{a,a'} &= \langle y_a, y_{a'} \rangle_{\mathcal{H}} \\ (h_Y)_a &= \langle \hat{\xi}, y_a \rangle_{\mathcal{H}}. \end{aligned}$$

Proof. First, $B_{XY} = Z_X Z_Y^*$, $G_{YY} = Z_Y Z_Y^*$, and $h_Y = Z_Y \hat{\xi}$. For example, (7.10) agrees with

$$\begin{aligned} Z_X Z_Y^* &= \left[\sum_{b=1}^n \sum_{i=1}^d e_{(b,i)} \otimes \partial_i k(X_b, \cdot) \right] \left[\sum_{a=1}^m y_a \otimes e_a \right] \\ &= \sum_{b=1}^n \sum_{i=1}^d \sum_{a=1}^m \langle \partial_i k(X_b, \cdot), y_a \rangle_{\mathcal{H}} \left[e_{(b,i)} \otimes e_a \right]. \end{aligned}$$

Recall the full-rank factorisation of pseudo-inverses: if a matrix A of rank r can be written as $A = FG$ for F, G each of rank r, then $A^\dagger = G^\dagger F^\dagger$ [18, Chapter 1, Section 6, Exercise 17].

Now we can show that the claimed form (7.9) matches $f_{\lambda,n}^m$ from (7.8):

$$\begin{aligned}
-Z_Y^*\left(\tfrac{1}{n}B_{XY}^{\mathsf{T}}B_{XY}+\lambda G_{YY}\right)^{\dagger}h_Y &= -Z_Y^*\left(\tfrac{1}{n}Z_YZ_X^*Z_XZ_Y^*+\lambda Z_YZ_Y^*\right)^{\dagger}Z_Y\hat{\xi} \\
&= -Z_Y^*\left(Z_Y\hat{C}_\lambda Z_Y^*\right)^{\dagger}Z_Y\hat{\xi} \\
&= -V\Sigma U^*\left((U\Sigma)(V^*\hat{C}_\lambda V)(\Sigma U^*)\right)^{\dagger}U\Sigma V^*\hat{\xi} \\
&= -V\Sigma U^*(\Sigma U^*)^{\dagger}(V^*\hat{C}_\lambda V)^{\dagger}(U\Sigma)^{\dagger}U\Sigma V^*\hat{\xi} \\
&= -V\Sigma U^*U\Sigma^{-1}(V^*\hat{C}_\lambda V)^{-1}\Sigma^{-1}U^*U\Sigma V^*\hat{\xi} \\
&= -V(V^*\hat{C}_\lambda V)^{-1}V^*\hat{\xi} \\
&= -g_Y(\hat{C}_\lambda)\hat{\xi} = f_{\lambda,n}^m. \qquad \square
\end{aligned}$$

Theorem 2 is the specialisation of Lemma 4 to $y_{(a,i)} = \partial_i k(Y_a,\cdot)$.

7.3 Consistency and convergence rates

We now address Theorem 3. To prove the consistency and convergence of $f_{\lambda,n}^m$, we first bound the difference between $f_{\lambda,n}^m$ in terms of various quantities, Section 7.3.1, which we then study individually in Section 7.3.2 to yield the final result in Section 7.3.3. Section 7.4 gives auxiliary results used along the way.

7.3.1 Decomposition

We care both about the parameter convergence $\|f_{\lambda,n}^m - f_0\|_{\mathcal{H}}$ and the convergence of $p_{\lambda,n}^m = p_{f_{\lambda,n}^m}$ to p_0 in various distances. By Lemma 1, we know that $J(p_0\|p_{\lambda,n}^m) = \frac{1}{2}\left\|C^{\frac{1}{2}}(f_{\lambda,n}^m - f_0)\right\|_{\mathcal{H}}^2$. In Lemma 16, we will additionally show that the L^r, KL, and Hellinger distances between the distributions can be bounded in terms of $\|f_{\lambda,n}^m - f_0\|_{\mathcal{H}}$. Thus it suffices to bound $\|C^{\alpha}(f_{\lambda,n}^m - f_0)\|_{\mathcal{H}}$ for $\alpha \geq 0$.

Lemma 5. *Under Assumptions (**A**) to (**F**), let $\alpha \geq 0$ and define*

$$c(a) := \lambda^{\min\left(0,\, a-\frac{1}{2}\right)}\|C\|^{\max\left(0,\, a-\frac{1}{2}\right)}, \qquad \mathcal{C}_Y := \|C_\lambda^{\frac{1}{2}}(I - VV^*)\|^2.$$

Then

$$\|C^{\alpha}(f^m_{\lambda,n}-f_0)\|_{\mathcal{H}} \leq R\,(2\mathcal{C}_Y+\lambda)\,c(\alpha)c(\beta)$$
$$+\frac{1}{\sqrt{\lambda}}\left\|C^{\alpha}\hat{C}_{\lambda}^{-\frac{1}{2}}\right\|\left(\|\hat{\xi}-\xi\|_{\mathcal{H}}+\|\hat{C}-C\|R\left((2\mathcal{C}_Y+\lambda)\,c(\alpha)c(\beta)+\|C\|^{\beta}\right)\right).$$

Proof. We decompose the error with respect to the best estimator for a fixed basis:

$$\begin{aligned} f^m_{\lambda} &:= \operatorname*{argmin}_{f\in\mathcal{H}_Y}\frac{1}{2}\langle f,P_YCP_Yf\rangle_{\mathcal{H}}+\langle f,P_Y\xi\rangle_{\mathcal{H}}+\frac{1}{2}\lambda\|f\|^2_{\mathcal{H}} \\ &= -(P_YCP_Y+\lambda I)^{-1}P_y\xi = -g_Y(C_\lambda)\xi = g_Y(C_\lambda)Cf_0. \end{aligned}$$

Then we have

$$\|C^{\alpha}(f^m_{\lambda,n}-f_0)\|_{\mathcal{H}} \leq \underbrace{\|C^{\alpha}(f^m_{\lambda,n}-f^m_{\lambda})\|_{\mathcal{H}}}_{\text{Estimation error}}+\underbrace{\|C^{\alpha}(f^m_{\lambda}-f_0)\|_{\mathcal{H}}}_{\text{Approximation error}}. \tag{7.11}$$

Approximation error

We consider the second term of (7.11) first. This term covers both approximation due to the basis $\mathcal{H}_Y$ and the bias due to regularisation. We break it down using ideas from the proof of Rudi et al. [99, Theorem 2]:

$$\begin{aligned} f_0-f^m_{\lambda} &= (I-g_Y(C_\lambda)C)f_0 \\ &= (I-g_Y(C_\lambda)C_\lambda+\lambda g_Y(C_\lambda))\,f_0 \\ &= (I-g_Y(C_\lambda)C_\lambda(VV^*)-g_Y(C_\lambda)C_\lambda(I-VV^*)+\lambda g_Y(C_\lambda))\,f_0 \\ &= ((I-VV^*)-g_Y(C_\lambda)C_\lambda(I-VV^*)+\lambda g_Y(C_\lambda))\,f_0, \end{aligned}$$

where in the last line we used Lemma 2 (iii). Thus, using Assumption (**F**) and Lemma 2 (v),

$$\begin{aligned}\|C^\alpha(f_\lambda^m - f_0)\|_{\mathcal{H}} &\le \|C^\alpha(I - VV^*)f_0\|_{\mathcal{H}} + \|C^\alpha g_Y(C_\lambda)C_\lambda(I - VV^*)f_0\|_{\mathcal{H}} \\ &\quad + \lambda\,\|C^\alpha g_Y(C_\lambda)f_0\|_{\mathcal{H}} \\ &\le \underbrace{\left\|C^\alpha C_\lambda^{-\frac{1}{2}}\right\|}_{\mathcal{S}_\alpha} \left\|C_\lambda^{\frac{1}{2}}(I - VV^*)C^\beta\right\| \underbrace{\left\|C^{-\beta}f_0\right\|_{\mathcal{H}}}_{\le R} \\ &\quad + \underbrace{\left\|C^\alpha C_\lambda^{-\frac{1}{2}}\right\|}_{\mathcal{S}_\alpha} \underbrace{\left\|C_\lambda^{\frac{1}{2}} g_Y(C_\lambda) C_\lambda^{\frac{1}{2}}\right\|}_{\le 1} \left\|C_\lambda^{\frac{1}{2}}(I - VV^*)C^\beta\right\| \underbrace{\left\|C^{-\beta}f_0\right\|_{\mathcal{H}}}_{\le R} \\ &\quad + \lambda \underbrace{\left\|C^\alpha C_\lambda^{-\frac{1}{2}}\right\|}_{\mathcal{S}_\alpha} \underbrace{\left\|C_\lambda^{\frac{1}{2}} g_Y(C_\lambda) C_\lambda^{\frac{1}{2}}\right\|}_{\le 1} \underbrace{\left\|C_\lambda^{-\frac{1}{2}}C^\beta\right\|_{\mathcal{H}}}_{\mathcal{S}_\beta} \underbrace{\left\|C^{-\beta}f_0\right\|_{\mathcal{H}}}_{\le R}.\end{aligned}$$

Because $P_Y = (I - VV^*)$ is a projection, we have

$$\left\|C_\lambda^{\frac{1}{2}}(I - VV^*)C^\beta\right\| \le \left\|C_\lambda^{\frac{1}{2}}(I - VV^*)^2 C_\lambda^{\frac{1}{2}}\right\| \left\|C_\lambda^{-\frac{1}{2}}C^\beta\right\| \le \left\|C_\lambda^{\frac{1}{2}}(I - VV^*)\right\|^2 \mathcal{S}_\beta.$$

We can also bound the terms $\mathcal{S}_a$ as follows. When $a \ge \frac{1}{2}$, the function $x \mapsto x^a/\sqrt{x+\lambda}$ is increasing on $[0,\infty)$, so that

$$\mathcal{S}_a = \left\|C_\lambda^{-\frac{1}{2}}C^a\right\| \le \frac{\|C\|^a}{\sqrt{\|C\| + \lambda}} \le \|C\|^{a-\frac{1}{2}}.$$

When instead $0 \le a < \frac{1}{2}$, since H is Hilbert-Schmidt, we have that

$$\mathcal{S}_a = \left\|C_\lambda^{-\frac{1}{2}}C^a\right\| \le \max_{x \ge 0} \frac{x^a}{\sqrt{x+\lambda}} = \sqrt{2}a^a\left(\frac{1}{2} - a\right)^{\frac{1}{2}-a} \lambda^{a-\frac{1}{2}} \le \lambda^{a-\frac{1}{2}}.$$

Combining the two yields

$$\mathcal{S}_a \le \lambda^{\min\left(0,\, a-\frac{1}{2}\right)} \|C\|^{\max\left(0,\, a-\frac{1}{2}\right)} = c(a),$$

and so

$$\|C^{\alpha}(f^m_{\lambda} - f_0)\|_{\mathcal{H}} \leq R\left(2\left\|C^{\frac{1}{2}}_{\lambda}(I - VV^*)\right\|^2 + \lambda\right)c(\alpha)c(\beta). \tag{7.12}$$

Estimation error

We continue with the first term of (7.11). Let $D = P_Y C P_Y$, $\hat{D} = P_Y \hat{C} P_Y$. Then

$$f^m_{\lambda} = -(D+\lambda I)^{-1}P_Y\xi = -\frac{1}{\lambda}(D+\lambda I - D)(D+\lambda I)^{-1}P_Y\xi = -\frac{1}{\lambda}(P_Y\xi + Df^m_{\lambda}),$$

and so the error due to finite n is

$$\begin{aligned} f^m_{\lambda} - f^m_{\lambda,n} &= (\hat{D}+\lambda I)^{-1}P_Y\hat{\xi} + f^m_{\lambda} \\ &= (\hat{D}+\lambda I)^{-1}\left(P_Y\hat{\xi} + (\hat{D}+\lambda I)f^m_{\lambda}\right) \\ &= (\hat{D}+\lambda I)^{-1}\left(P_Y\hat{\xi} + \hat{D}f^m_{\lambda} + \lambda f^m_{\lambda}\right) \\ &= (\hat{D}+\lambda I)^{-1}\left(P_Y\hat{\xi} + \hat{D}f^m_{\lambda} - P_Y\xi - Df^m_{\lambda}\right) \\ &= (\hat{D}+\lambda I)^{-1}\left(P_Y(\hat{\xi}-\xi) + (\hat{D}-D)f^m_{\lambda}\right) \\ &= (\hat{D}+\lambda I)^{-1}\left(P_Y(\hat{\xi}-\xi) + (\hat{D}-D)(f^m_{\lambda}-f_0) + (\hat{D}-D)f_0\right). \end{aligned}$$

We thus have, using $\|P_Y\| \leq 1$,

$$\begin{aligned} \|C^{\alpha}(f^m_{\lambda} - f^m_{\lambda,n})\|_{\mathcal{H}} \leq \left\|C^{\alpha}(P_Y\hat{C}P_Y + \lambda I)^{-1}P_Y\right\| \Big(& \\ \|\hat{\xi}-\xi\|_{\mathcal{H}} + \|\hat{C}-C\|\|f^m_{\lambda}-f_0\|_{\mathcal{H}} + \|\hat{C}-C\|\,\|f_0\|_{\mathcal{H}}\Big). & \end{aligned}$$

We have already bounded $\|f^m_{\lambda} - f_0\|_{\mathcal{H}}$, and have $\|f_0\|_{\mathcal{H}} \leq \|C^{\beta}\|\|C^{-\beta}f_0\|_{\mathcal{H}} \leq R\|C\|^{\beta}$. Using Lemma 2 (iv) and (v), we have

$$\begin{aligned} \left\|C^{\alpha}(P_Y\hat{C}P_Y + \lambda I)^{-1}P_Y\right\| = \|C^{\alpha}g_Y(\hat{C}_{\lambda})\| &\leq \left\|C^{\alpha}\hat{C}_{\lambda}^{-\frac{1}{2}}\right\|\left\|\hat{C}_{\lambda}^{\frac{1}{2}}g_Y(\hat{C}_{\lambda})\hat{C}_{\lambda}^{\frac{1}{2}}\right\|\left\|\hat{C}_{\lambda}^{-\frac{1}{2}}\right\| \\ &\leq \frac{1}{\sqrt{\lambda}}\left\|C^{\alpha}\hat{C}_{\lambda}^{-\frac{1}{2}}\right\|, \end{aligned}$$

and so $\left\|C^{\alpha}(f^m_{\lambda} - f^m_{\lambda,n})\right\|_{\mathcal{H}}$ is upper bounded by

$$\frac{\left\|C^{\alpha}\hat{C}_{\lambda}^{-\frac{1}{2}}\right\|}{\sqrt{\lambda}}\left(\|\hat{\xi} - \xi\|_{\mathcal{H}} + \|\hat{C} - C\|\left(\|f^m_{\lambda} - f_0\|_{\mathcal{H}} + R\|C\|^{\beta}\right)\right). \quad (7.13)$$

The claim follows by using (7.12) and (7.13) in (7.11). □

7.3.2 Probabilistic inequalities

We only need Lemma 5 for $\alpha = 0$ and $\alpha = \frac{1}{2}$; in the former case, we use $\left\|\hat{C}_{\lambda}^{-\frac{1}{2}}\right\| \leq 1/\sqrt{\lambda}$. Thus we are left with four quantities to control: $\|C^{\frac{1}{2}}\hat{C}_{\lambda}^{-\frac{1}{2}}\|$, $\mathcal{C}_Y = \|C_{\lambda}^{\frac{1}{2}}(I - VV^*)\|^2$, $\|\hat{\xi} - \xi\|_{\mathcal{H}}$, and $\|\hat{C} - C\|$.

Lemma 6. *Let $\rho, \delta \in (0,1)$. Under Assumptions (**B**) to (**E**) and (**G**) for any $0 < \lambda \leq \frac{1}{4}\|C\|$, define $w := \log\frac{25 \operatorname{Tr} C}{\lambda\delta}$. When*

$$n \geq \max\left(\frac{4w}{3\rho}, \frac{40d\mathcal{N}'_{\infty}(\lambda)w}{\rho^2}\right),$$

we have that with probability at least $1 - \delta$,

$$\|C^{\frac{1}{2}}\hat{C}_{\lambda}^{-\frac{1}{2}}\| \leq \frac{1}{\sqrt{1-\rho}}.$$

Proof. Let $\gamma := \lambda_{\max}\left(C_{\lambda}^{-\frac{1}{2}}(C - \hat{C})C_{\lambda}^{-\frac{1}{2}}\right)$. Lemma 15 gives that $\|C^{\frac{1}{2}}\hat{C}_{\lambda}^{-\frac{1}{2}}\| \leq \frac{1}{\sqrt{1-\gamma}}$ as long as $\gamma < 1$. We bound γ with Lemma 13, using $Y^a_i = \partial_i k(X_a, \cdot)$ so that $\mathbb{E}\sum_{i=1}^{d} Y^a_i \otimes Y^a_i = C$. This gives us that $\gamma \leq \rho$ with probability at least $1 - \delta$ as long as

$$\rho \leq \frac{2w}{3n} + \sqrt{\frac{10d\mathcal{N}'_{\infty}(\lambda)w}{n}},$$

which is satisfied by the condition on n. □

Lemma 7. *Choose the basis $y_{(a,i)} = \partial_i k(Y_a, \cdot)$, with the points $\{Y_a\}_{a\in[m]}$ sampled i.i.d. from p_0. Let $\rho, \delta \in (0,1)$, and define $w := \log\frac{25 \operatorname{Tr}(C)}{\lambda\delta}$. Then, under*

*Assumptions (**B**) to (**E**) and (**G**)*

$$\mathcal{C}_Y = \|C_\lambda^{\frac{1}{2}}(I - VV^*)\|^2 \leq \frac{\lambda}{1-\rho}$$

with probability at least $1-\delta$ *as long as*

$$m \geq \max\left(\frac{4w}{3\rho}, \frac{40d\mathcal{N}'_\infty(\lambda)w}{\rho^2}\right).$$

Proof. We have that $\|C_\lambda^{\frac{1}{2}}(I - VV^*)\|^2 \leq \lambda \left\|(\frac{1}{m}Z_Y^* Z_Y + \lambda I)^{-\frac{1}{2}} C_\lambda^{\frac{1}{2}}\right\|^2$ via Lemma 14. Again applying Lemmas 13 and 15 as in the proof of Lemma 6 yields the result. □

For the remaining two quantities, we use simple Hoeffding[1] bounds.

Lemma 8 (Concentration of $\hat{\xi}$)**.** *Under Assumption (**G**), with probability at least* $1-\delta$ *we have*

$$\|\hat{\xi} - \xi\|_{\mathcal{H}} \leq \frac{2d(Q\kappa_1 + \kappa_2)}{\sqrt{n}}\left(1 + \sqrt{2\log\frac{1}{\delta}}\right).$$

Proof. Let

$$\nu_a := \sum_{i=1}^d \left(\partial_i \log q_0(X_a)\partial_i k(X_a, \cdot) + \partial_i^2 k(X_a, \cdot)\right) - \xi,$$

so that $\hat{\xi} - \xi = \frac{1}{n}\sum_{a=1}^n \nu_a$, and for each a we have that $\mathbb{E}\,\nu_a = 0$ and

$$\|\nu_a\|_{\mathcal{H}} \leq 2\sup_{x\in\Omega}\left\|\sum_{i=1}^d \partial_i \log q_0(x)\partial_i k(x, \cdot) + \partial_i^2 k(x, \cdot)\right\|_{\mathcal{H}} \leq 2d\left(Q\kappa_1 + \kappa_2\right).$$

Applying Lemma 10 to the vectors ν_a gives the result. □

Lemma 9 (Concentration of $\hat{C}$)**.** *Under Assumption (**G**), with probability at*

[1] A Bernstein bound would allow for a slightly better result when κ_1 and κ_2 are large, at the cost of a more complex form.

least $1-\delta$ *we have*

$$\|\hat{C}-C\| \le \frac{2d\kappa_1^2}{\sqrt{n}}\left(1+\sqrt{2\log\tfrac{1}{\delta}}\right).$$

Proof. Let

$$C_x := \sum_{i=1}^{d} \partial_i k(x,\cdot) \otimes \partial_i k(x,\cdot),$$

so that $\hat{C} = \frac{1}{n}\sum_{a=1} n C_{X_a}$, $C = \mathbb{E}\, C_x$. We know that

$$\begin{aligned}\|C_x - C\| &\le 2\sum_{i=1}^{d} \|\partial_i k(x,\cdot)\|_{\mathcal{H}}^2 \le 2d\kappa_1^2 \\ \|C_x - C\|_{\text{HS}} &\le 2\sum_{i=1}^{d} \sup_{x\in\Omega} \|\partial_i k(x,\cdot)\|_{\mathcal{H}}^2 \le 2d\kappa_1^2,\end{aligned}$$

so applying Lemma 11 shows the result. □

7.3.3 Final bound

Theorem 4 (Finite-sample convergence of $f_{\lambda,n}^m$)**.** *Under Assumptions (**A**) to (**G**), let* $\delta \in (0,1)$ *and define* $S_\delta := 1 + \sqrt{2\log\frac{4}{\delta}}$*. Use the basis* $y_{(a,i)} = \partial_i k(Y_a,\cdot)$*, for* $\{Y_a\}_{a=1}^m$ *an iid sample from* p_0 *not necessarily independent of* X*. Assume that* $0 < \lambda < \frac{1}{4}\|C\|$*. When*

$$\min(n,m) \ge \frac{90d\kappa_1^2}{\lambda}\log\frac{100d\kappa_1^2}{\lambda\delta},$$

we have with probability at least $1-\delta$ *that both of the following hold simultaneously:*

$$\begin{aligned}
\|f^m_{\lambda,n} - f_0\|_{\mathcal{H}} \le{}& 7R\lambda^{\min\left(\frac{1}{2},\,\beta\right)}(d\kappa_1^2)^{\max\left(0,\,\beta-\frac{1}{2}\right)} \\
&+ \frac{2d}{\lambda\sqrt{n}} S_\delta\Big(Q\kappa_1 + \kappa_2 \\
&\qquad + R\kappa_1^2\Big(7\lambda^{\min\left(\frac{1}{2},\,\beta\right)}(d\kappa_1^2)^{\max\left(0,\,\beta-\frac{1}{2}\right)} + (d\kappa_1^2)^{\beta}\Big)\Big) \\
\|C^{\frac{1}{2}}(f^m_{\lambda,n} - f_0)\|_{\mathcal{H}} \le{}& 7R\lambda^{\min\left(1,\,\beta+\frac{1}{2}\right)}(d\kappa_1^2)^{\max\left(0,\,\beta-\frac{1}{2}\right)} \\
&+ \frac{2d\sqrt{3}}{\sqrt{\lambda n}} S_\delta\Big(Q\kappa_1 + \kappa_2 \\
&\qquad + R\kappa_1^2\Big(7\lambda^{\min\left(\frac{1}{2},\,\beta\right)}(d\kappa_1^2)^{\max\left(0,\,\beta-\frac{1}{2}\right)} + (d\kappa_1^2)^{\beta}\Big)\Big).
\end{aligned}$$

Proof. Recall from Lemma 5 that

$$\begin{aligned}
&\|C^{\alpha}(f^m_{\lambda,n} - f_0)\|_{\mathcal{H}} \le R\,(2\mathcal{C}_Y + \lambda)\,c(\alpha)c(\beta) \\
&+ \frac{1}{\sqrt{\lambda}}\left\|C^{\alpha}\hat{C}_{\lambda}^{-\frac{1}{2}}\right\|\Big(\|\hat{\xi} - \xi\|_{\mathcal{H}} + \|\hat{C} - C\|R\Big((2\mathcal{C}_Y + \lambda)\,c(\alpha)c(\beta) + \|C\|^{\beta}\Big)\Big).
\end{aligned}$$

We use a union bound over the results of Lemmas 6 to 9. Note that under Assumption (**G**), each of $\|C\|$ and $\operatorname{Tr} C$ are at most $d\kappa_1^2$ and $\mathcal{N}'_{\infty}(\lambda) \le \kappa_1^2/\lambda$.

We first use $\rho = \frac{2}{3}$ in Lemmas 6 and 7 to get that $\mathcal{C}_Y \le 3\lambda$ and $\|C^{\frac{1}{2}}\hat{C}_{\lambda}^{-\frac{1}{2}}\| \le \sqrt{3}$ with probability at least $\frac{\delta}{2}$ when n and m are each at least

$$\max\left(2,\, 90d\mathcal{N}'_{\infty}(\lambda)\right)\log\frac{25\operatorname{Tr} C}{\lambda\frac{\delta}{4}} \le \frac{90d\kappa_1^2}{\lambda}\log\frac{100d\kappa_1^2}{\lambda\delta},$$

where we used that $\lambda < \frac{1}{4}\|C\|$. The claim follows from applying Lemmas 8 and 9. □

Theorem 3 now follows from considering the asymptotics of Theorem 4, once we additionally make Assumption (**H**):

Proof of Theorem 3. Let $b := \min\left(\frac{1}{2},\, \beta\right)$. Under Assumptions (**A**) to (**G**), as $n \to \infty$ Theorem 4 gives:

$$\begin{aligned}
\|f^m_{\lambda,n} - f_0\|_{\mathcal{H}} &= \mathcal{O}_{p_0}\left(\lambda^b + n^{-\frac{1}{2}}\lambda^{-1} + n^{-\frac{1}{2}}\lambda^{b-1}\right) \\
&= \mathcal{O}_{p_0}\left(\lambda^b + n^{-\frac{1}{2}}\lambda^{-1}\right) \\
\|C^{\frac{1}{2}}(f^m_{\lambda,n} - f_0)\|_{\mathcal{H}} &= \mathcal{O}_{p_0}\left(\lambda^{b+\frac{1}{2}} + n^{-\frac{1}{2}}\lambda^{-\frac{1}{2}} + n^{-\frac{1}{2}}\lambda^{b-\frac{1}{2}}\right) \\
&= \mathcal{O}_{p_0}\left(\lambda^{b+\frac{1}{2}} + n^{-\frac{1}{2}}\lambda^{-\frac{1}{2}}\right)
\end{aligned}$$

as long as $\min(n,m) = \Omega(\lambda^{-1}\log\lambda^{-1})$. Choosing $\lambda = n^{-\theta}$, this requirement is $\min(n,m) = \Omega(n^{\theta}\log n)$ and the bounds become

$$\begin{aligned}
\|f^m_{\lambda,n} - f_0\|_{\mathcal{H}} &= \mathcal{O}_{p_0}\left(n^{-b\theta} + n^{\theta-\frac{1}{2}}\right) \\
\|C^{\frac{1}{2}}(f^m_{\lambda,n} - f_0)\|_{\mathcal{H}} &= \mathcal{O}_{p_0}\left(n^{-b\theta-\frac{1}{2}\theta} + n^{\frac{1}{2}\theta-\frac{1}{2}}\right).
\end{aligned}$$

Both bounds are minimised when $\theta = \frac{1}{2(1+b)}$, which since $0 \le b \le \frac{1}{2}$ leads to $\frac{1}{2} \ge \theta \ge \frac{1}{3}$, and the requirement on n is always satisfied once n is large enough. This shows, as claimed, that

$$\|f^m_{\lambda,n} - f_0\|_{\mathcal{H}} = \mathcal{O}_{p_0}\left(n^{-\frac{b}{2(b+1)}}\right) \qquad J(p_0\|p_{f^m_{\lambda,n}}) = \mathcal{O}_{p_0}\left(n^{-\frac{2b+1}{2(b+1)}}\right)$$

when $m = \Omega\left(n^{\frac{1}{2(1+b)}}\log n\right)$.

The bounds on L^r, Hellinger, and KL convergence follow from Lemma 16 under Assumption (**H**). □

7.4 Auxiliary results

7.4.1 Concentration inequalities in Hilbert spaces

Lemma 10 (Hoeffding-type inequality for random vectors). *Let $X_1, \dots, X_n$ be iid random variables in a (separable) Hilbert space, where $\mathbb{E}\, X_i = 0$ and $\|X_i\| \le L$*

almost surely. Then for any $\varepsilon > L/\sqrt{n}$*,*

$$\Pr\left(\left\|\frac{1}{n}\sum_{i=1}^{n} X_i\right\| > \varepsilon\right) \le \exp\left(-\frac{1}{2}\left(\frac{\sqrt{n}\varepsilon}{L} - 1\right)^2\right);$$

equivalently, we have with probability at least $1 - \delta$ *that*

$$\left\|\frac{1}{n}\sum_{i=1}^{n} X_i\right\| \le \frac{L}{\sqrt{n}}\left(1 + \sqrt{2\log\tfrac{1}{\delta}}\right).$$

Proof. Following Boucheron et al. [23, Example 6.3], we can apply McDiarmid's inequality. The function $f(X_1, \dots, X_n) = \left\|\frac{1}{n}\sum_{i=1}^{n} X_i\right\|$ satisfies bounded differences:

$$\left|\left\|\frac{1}{n}\sum_{i=1}^{n} X_i\right\| - \left\|\frac{1}{n}\hat{X}_1 + \frac{1}{n}\sum_{i=2}^{n} X_i\right\|\right| \le \left\|\frac{1}{n}(X_1 - \hat{X}_1)\right\| \le \frac{2L}{n}.$$

Thus for $\varepsilon \ge \mathbb{E}\left\|\frac{1}{n}\sum_i X_i\right\|$,

$$\Pr\left(\left\|\frac{1}{n}\sum_{i} X_i\right\| > \varepsilon\right) \le \exp\left(-\frac{n\left(\varepsilon - \mathbb{E}\left\|\frac{1}{n}\sum_i X_i\right\|\right)^2}{2L^2}\right).$$

We also know that

$$\begin{aligned}\mathbb{E}\left\|\frac{1}{n}\sum_i X_i\right\| &\le \frac{1}{n}\sqrt{\mathbb{E}\left\|\sum_i X_i\right\|^2} = \frac{1}{n}\sqrt{\sum_{i,j}\mathbb{E}\langle X_i, X_j\rangle} = \frac{1}{n}\sqrt{\sum_i \mathbb{E}\|X_i\|^2} \\ &\le \frac{1}{n}\sqrt{nL^2} = \frac{L}{\sqrt{n}},\end{aligned}$$

so

$$\Pr\left(\left\|\frac{1}{n}\sum_i X_i\right\| > \varepsilon\right) \le \exp\left(-\frac{n\left(\varepsilon - \frac{L}{\sqrt{n}}\right)^2}{2L^2}\right) = \exp\left(-\frac{1}{2}\left(\frac{\sqrt{n}\varepsilon}{L} - 1\right)^2\right)$$

as desired. The second statement follows by simple algebra. □

Lemma 11 (Hoeffding-type inequality for random Hilbert-Schmidt operators)**.** *Let $X_1, \ldots, X_n$ be iid random operators in a (separable) Hilbert space, where $\mathbb{E} X_i = 0$ and $\|X_i\| \le L$, $\|X_i\|_{\mathrm{HS}} \le B$ almost surely. Then for any $\varepsilon > B/\sqrt{n}$,*

$$\Pr\left(\left\| \frac{1}{n} \sum_{i=1}^{n} X_i \right\| < \varepsilon \right) \le \exp\left(-\frac{1}{2} \left(\frac{\sqrt{n}\varepsilon}{L} - \frac{B}{L} \right)^2 \right);$$

equivalently, we have with probability at least $1 - \delta$ that

$$\left\| \frac{1}{n} \sum_{i=1}^{n} X_i \right\| \le \frac{1}{\sqrt{n}} \left(B + L\sqrt{2 \log \tfrac{1}{\delta}} \right).$$

Proof. The argument is the same as Lemma 10, except that

$$\begin{aligned}\mathbb{E} \left\| \frac{1}{n} \sum_i X_i \right\| &\le \frac{1}{n} \sqrt{ \mathbb{E} \left\| \sum_i X_i \right\|_{\mathrm{HS}}^2 } = \frac{1}{n} \sqrt{ \sum_{i,j} \mathbb{E} \langle X_i, X_j \rangle_{\mathrm{HS}} } = \frac{1}{n} \sqrt{ \sum_i \mathbb{E} \|X_i\|_{\mathrm{HS}}^2 } \\ &\le \frac{B}{\sqrt{n}}\end{aligned}$$

using $\|X_i\| \le \|X_i\|_{\mathrm{HS}}$. □

Lemma 12 (Bernstein's inequality for a sum of random operators; Proposition 12 of Rudi et al. [99])**.** *Let $\mathcal{H}$ be a separable Hilbert space, and $X_1, \ldots, X_n$ a sequence of iid self-adjoint positive random operators on $\mathcal{H}$, with $\mathbb{E} X_1 = 0$, $\lambda_{\max}(X_1) \le L$ almost surely for some $L > 0$. Let S be a positive operator such that $\mathbb{E}[X_1^2] \preceq S$. Let $\beta = \log \frac{2 \operatorname{Tr} S}{\|S\| \delta}$. Then for any $\delta \ge 0$, with probability at least $1 - \delta$*

$$\lambda_{\max}\left(\frac{1}{n} \sum_{i=1}^{n} X_i \right) \le \frac{2L\beta}{3n} + \sqrt{\frac{2\|S\|\beta}{n}}.$$

7.4.2 Concentration of sum of correlated operators

The following result is similar to Rudi et al. [99, Proposition 8], but the proof is considerably more complex due to the sum over correlated operators.

Lemma 13. *Let $W_a = (Y_i^a)_{i\in[d]}$ be a random d-tuple of vectors in a separable Hilbert space $\mathcal{H}$, with $\{W_a\}_{a\in[n]}$ iid. Suppose that $Q = \mathbb{E}\sum_{i=1}^d Y_i^1 \otimes Y_i^1$ exists and is trace class, and that for any $\lambda > 0$ there is $\mathcal{N}'_\infty(\lambda) < \infty$ such that $\langle Y_l^a, (Q + \lambda I)^{-1} Y_l^a\rangle_{\mathcal{H}} \le \mathcal{N}'_\infty(\lambda)$ almost surely. Let $Q_\lambda = Q + \lambda I$, $V_a = \sum_{i=1}^d Y_i^a \otimes Y_i^a$.*

For any $0 < \rho < \frac{1}{2}$ and any $0 < \lambda \le \rho\|Q\|$, for any $\delta \ge 0$ it holds with probability at least $1 - \delta$ that

$$\lambda_{\max}\left(Q_\lambda^{-\frac{1}{2}}\left(Q - \frac{1}{n}\sum_{a=1}^n V_a\right)Q_\lambda^{-\frac{1}{2}}\right) \le \frac{2\beta}{3n} + \sqrt{\frac{10d\mathcal{N}'(\lambda)\beta}{n}},$$

with $\beta = \log\left(\frac{10\,\mathrm{Tr}\,Q}{\lambda\delta\left(\frac{3}{1+\rho}-2\right)}\right)$.

Proof. We apply the Bernstein inequality for random operators, Lemma 12, to $Z_a := Q_\lambda^{-\frac{1}{2}}(Q - V_a)Q_\lambda^{-\frac{1}{2}}$. For each a, clearly $\mathbb{E}\,Z_a = 0$, and since V_a is positive

$$\begin{aligned}\sup_{\|f\|_{\mathcal{H}}=1} \langle f, Z_a f\rangle_{\mathcal{H}} &= \sup_{\|f\|_{\mathcal{H}}=1} \langle f, Q_\lambda^{-1} Q f\rangle_{\mathcal{H}} - \langle f, Q_\lambda^{-\frac{1}{2}} V_a Q_\lambda^{-\frac{1}{2}} f\rangle_{\mathcal{H}} \\ &\le \sup_{\|f\|_{\mathcal{H}}=1} \langle f, Q_\lambda^{-1} Q f\rangle_{\mathcal{H}} \\ &\le 1.\end{aligned}$$

To apply Lemma 12, we now need to find a positive operator S to upper bound the second moment of Z_a. Letting $u \in \mathcal{H}$, and dropping the dependence on a for brevity, we have that

$$\begin{aligned}\langle u,\mathbb{E}[Z^2]u\rangle_{\mathcal{H}} &= \left\langle u,\mathbb{E}[Q_\lambda^{-\frac{1}{2}}VQ_\lambda^{-1}VQ_\lambda^{-\frac{1}{2}}]u\right\rangle_{\mathcal{H}} - \left\langle u,Q_\lambda^{-\frac{1}{2}}QQ_\lambda^{-1}QQ_\lambda^{-\frac{1}{2}}u\right\rangle_{\mathcal{H}} \\ &\leq \left\langle u,Q_\lambda^{-\frac{1}{2}}\,\mathbb{E}[VQ_\lambda^{-1}V]Q_\lambda^{-\frac{1}{2}}u\right\rangle_{\mathcal{H}} \\ &= \left\langle Q_\lambda^{-\frac{1}{2}}u,\mathbb{E}[VQ_\lambda^{-1}V]Q_\lambda^{-\frac{1}{2}}u\right\rangle_{\mathcal{H}} \\ &= \sum_{i,j}^{d}\left\langle Q_\lambda^{-\frac{1}{2}}u,\mathbb{E}[(Y_i\otimes Y_i)Q_\lambda^{-1}(Y_j\otimes Y_j)]Q_\lambda^{-\frac{1}{2}}u\right\rangle_{\mathcal{H}} \\ &= \sum_{i,j}^{d}\mathbb{E}\left[\langle Q_\lambda^{-\frac{1}{2}}u,Y_i\rangle_{\mathcal{H}}\langle Q_\lambda^{-\frac{1}{2}}u,Y_j\rangle_{\mathcal{H}}\langle Y_i,Q_\lambda^{-1}Y_j\rangle_{\mathcal{H}}\right].\end{aligned}$$

Using the identity $2\langle x,Ay\rangle = \langle x+y,A(x+y)\rangle - \langle x,Ax\rangle - \langle y,Ay\rangle$, we get:

$$\begin{aligned}\langle u,\mathbb{E}[Z^2]u\rangle_{\mathcal{H}} \leq \frac{1}{2}\sum_{i,j}^{d}\mathbb{E}&\left[\langle Q_\lambda^{-\frac{1}{2}}u,Y_i\rangle_{\mathcal{H}}\langle Q_\lambda^{-\frac{1}{2}}u,Y_j\rangle_{\mathcal{H}}\langle Y_i+Y_j,Q_\lambda^{-1}(Y_i+Y_j)\rangle_{\mathcal{H}}\right] \\ &-\sum_{i,j}^{d}\mathbb{E}\left[\langle Q_\lambda^{-\frac{1}{2}}u,Y_i\rangle_{\mathcal{H}}\langle Q_\lambda^{-\frac{1}{2}}u,Y_j\rangle_{\mathcal{H}}\langle Y_i,Q_\lambda^{-1}Y_i\rangle\right].\end{aligned}$$

Similarly, using $2\langle A,x\rangle\langle A,y\rangle = \langle A,x+y\rangle^2 - \langle A,x\rangle^2 - \langle A,y\rangle^2$, we get that the first line is

$$\begin{aligned}\sum_{i,j}^{d}\frac{1}{4}\mathbb{E}&\left[\langle Q_\lambda^{-\frac{1}{2}}u,Y_i+Y_j\rangle_{\mathcal{H}}^2\langle Y_i+Y_j,Q_\lambda^{-1}(Y_i+Y_j)\rangle_{\mathcal{H}}\right] \\ &-\frac{1}{2}\mathbb{E}\left[\langle Q_\lambda^{-\frac{1}{2}}u,Y_i\rangle_{\mathcal{H}}^2\langle Y_i+Y_j,Q_\lambda^{-1}(Y_i+Y_j)\rangle_{\mathcal{H}}\right],\end{aligned}$$

and the second is

$$\begin{aligned}\frac{1}{2}\sum_{i,j}^{d}-\mathbb{E}&\left[\langle Q_\lambda^{-\frac{1}{2}}u,Y_i+Y_j\rangle_{\mathcal{H}}^2\langle Y_i,Q_\lambda^{-1}Y_i\rangle\right] \\ &+\mathbb{E}\left[\langle Q_\lambda^{-\frac{1}{2}}u,Y_i\rangle_{\mathcal{H}}^2\langle Y_i,Q_\lambda^{-1}Y_i\rangle\right] \\ &+\mathbb{E}\left[\langle Q_\lambda^{-\frac{1}{2}}u,Y_j\rangle_{\mathcal{H}}^2\langle Y_i,Q_\lambda^{-1}Y_i\rangle\right].\end{aligned}$$

Each of these expectations is non-negative, so dropping the ones with negative coefficients gives:

$$\begin{aligned}\langle u, \mathbb{E}[Z^2]u\rangle_{\mathcal{H}} \leq \frac{1}{4}\sum_{i,j}^{d} \mathbb{E}\left[\langle Q_\lambda^{-\frac{1}{2}}u, Y_i + Y_j\rangle_{\mathcal{H}}^2 \langle Y_i + Y_j, Q_\lambda^{-1}(Y_i + Y_j)\rangle_{\mathcal{H}}\right] \\ + \frac{1}{2}\sum_{i,j}^{d} \mathbb{E}\left[\langle Q_\lambda^{-\frac{1}{2}}u, Y_i\rangle_{\mathcal{H}}^2 \langle Y_i, Q_\lambda^{-1}Y_i\rangle\right] \\ + \frac{1}{2}\sum_{i,j}^{d} \mathbb{E}\left[\langle Q_\lambda^{-\frac{1}{2}}u, Y_j\rangle_{\mathcal{H}}^2 \langle Y_i, Q_\lambda^{-1}Y_i\rangle\right].\end{aligned}$$

Recalling that $\langle Y_i, Q_\lambda^{-1}Y_i\rangle \leq \mathcal{N}'_\infty(\lambda)$, the latter two sums are upper-bounded by $\mathcal{N}'_\infty(\lambda)$ times

$$\frac{1}{2}\sum_{i,j}^{d} \mathbb{E}\left[\langle Q_\lambda^{-\frac{1}{2}}u, Y_i\rangle_{\mathcal{H}}^2\right] + \frac{1}{2}\sum_{i,j}^{d} \mathbb{E}\left[\langle Q_\lambda^{-\frac{1}{2}}u, Y_j\rangle_{\mathcal{H}}^2\right] = d\sum_{i=1}^{d} \mathbb{E}\left[\langle Q_\lambda^{-\frac{1}{2}}u, Y_i\rangle_{\mathcal{H}}^2\right].$$

We also have that

$$\begin{aligned}\langle Y_i + Y_j, Q_\lambda^{-1}(Y_i + Y_j)\rangle_{\mathcal{H}} &= \|Q_\lambda^{-\frac{1}{2}}(Y_i + Y_j)\|_{\mathcal{H}}^2 \\ &\leq 2(\|Q_\lambda^{-\frac{1}{2}}Y_i\|_{\mathcal{H}}^2 + \|Q_\lambda^{-\frac{1}{2}}Y_i\|_{\mathcal{H}}^2) \\ &\leq 4\mathcal{N}'_\infty(\lambda),\end{aligned}$$

so the first sum is at most $\mathcal{N}'_\infty(\lambda)$ times

$$\begin{aligned}\sum_{i,j}^{d} \mathbb{E}\left[\langle Q_\lambda^{-\frac{1}{2}}u, Y_i + Y_j\rangle_{\mathcal{H}}^2\right] &= \sum_{i,j}^{d} \mathbb{E}\Big[\langle Q_\lambda^{-\frac{1}{2}}u, Y_i\rangle_{\mathcal{H}}^2 + \langle Q_\lambda^{-\frac{1}{2}}u, Y_j\rangle_{\mathcal{H}}^2 \\ &\qquad + 2\langle Q_\lambda^{-\frac{1}{2}}u, Y_i\rangle_{\mathcal{H}}\langle Q_\lambda^{-\frac{1}{2}}u, Y_j\rangle_{\mathcal{H}}\Big] \\ &= 2d\sum_{i=1}^{d} \mathbb{E}\left[\langle Q_\lambda^{-\frac{1}{2}}u, Y_i\rangle_{\mathcal{H}}^2\right] \\ &\quad + 2\sum_{i,j}^{d} \mathbb{E}\left[\langle Q_\lambda^{-\frac{1}{2}}u, Y_i\rangle_{\mathcal{H}}\langle Q_\lambda^{-\frac{1}{2}}u, Y_j\rangle_{\mathcal{H}}\right].\end{aligned}$$

Thus, $\langle u, \mathbb{E}[Z^2]u\rangle_{\mathcal{H}}$ is upper bounded by

$$\begin{aligned}
&\mathcal{N}'_\infty(\lambda)\left(2\sum_{i,j}^{d}\mathbb{E}\left[\langle Q_\lambda^{-\frac{1}{2}}u, Y_i\rangle_{\mathcal{H}}\langle Q_\lambda^{-\frac{1}{2}}u, Y_j\rangle_{\mathcal{H}}\right] + 3d\sum_{i=1}^{d}\mathbb{E}\left[\langle Q_\lambda^{-\frac{1}{2}}u, Y_i\rangle_{\mathcal{H}}^2\right]\right)\\
&= \left\langle u, \mathcal{N}'_\infty(\lambda)Q_\lambda^{-\frac{1}{2}}\left(2\mathbb{E}\left[\sum_{i,j}^{d}Y_i\otimes Y_j\right] + 3d\,\mathbb{E}\left[\sum_{i=1}^{d}Y_i\otimes Y_i\right]\right)Q_\lambda^{-\frac{1}{2}}u\right\rangle_{\mathcal{H}}\\
&= \left\langle u, \mathcal{N}'_\infty(\lambda)Q_\lambda^{-\frac{1}{2}}\left(2M+3dQ\right)Q_\lambda^{-\frac{1}{2}}u\right\rangle_{\mathcal{H}},
\end{aligned}$$

where we defined $M := \mathbb{E}\left[\left(\sum_{i=1}^{d}Y_i\right)\otimes\left(\sum_{i=1}^{d}Y_i\right)\right]$. Thus we have the desired upper bound $E[Z^2] \preceq S := \mathcal{N}'_\infty(\lambda)Q_\lambda^{-\frac{1}{2}}(2M+3dQ)Q_\lambda^{-\frac{1}{2}}$. In order to be able to finally use Lemma 12, we still need to find an upper bound on $\operatorname{Tr} S$, and both upper and lower bounds on $\|S\|$.

To do so, we first show that $M \preceq dQ$:

$$\begin{aligned}
\langle u, Mu\rangle_{\mathcal{H}} &= \left\langle u, \mathbb{E}\left[\left(\sum_{i=1}^{d}Y_i\right)\otimes\left(\sum_{i=1}^{d}Y_i\right)\right]u\right\rangle_{\mathcal{H}} = \mathbb{E}\left[\left\langle u, \sum_{i=1}^{d}Y_i\right\rangle_{\mathcal{H}}^2\right]\\
&\leq \mathbb{E}\left[d\sum_{i=1}^{d}\langle u, Y_i\rangle_{\mathcal{H}}^2\right] = \mathbb{E}\left[d\sum_{i=1}^{d}\langle u, (Y_i\otimes Y_i)u\rangle_{\mathcal{H}}\right] = \langle u, dQu\rangle_{\mathcal{H}}.
\end{aligned}$$

Thus $\operatorname{Tr}(M) \leq d\operatorname{Tr}(Q)$, and so

$$\begin{aligned}
\operatorname{Tr}(S) &= \mathcal{N}'_\infty(\lambda)\left(2\operatorname{Tr}(Q_\lambda^{-\frac{1}{2}}MQ_\lambda^{-\frac{1}{2}}) + 3d\operatorname{Tr}(Q_\lambda^{-\frac{1}{2}}QQ_\lambda^{-\frac{1}{2}})\right)\\
&= \mathcal{N}'_\infty(\lambda)\left(2\operatorname{Tr}(Q_\lambda^{-1}M) + 3d\operatorname{Tr}(Q_\lambda^{-1}Q\right)\\
&\leq \mathcal{N}'_\infty(\lambda)\left(\frac{2}{\lambda}\operatorname{Tr}(M) + \frac{3d}{\lambda}\operatorname{Tr}(Q)\right)\\
&\leq \frac{5d}{\lambda}\mathcal{N}'_\infty(\lambda)\operatorname{Tr}(Q).
\end{aligned}$$

We next bound $\|S\|$. Again because $M \preceq dQ$, we have that

$$\begin{aligned}\langle u, Q_\lambda^{-\frac{1}{2}} M Q_\lambda^{-\frac{1}{2}} u\rangle_{\mathcal{H}} &= \langle Q_\lambda^{-\frac{1}{2}} u, M(Q_\lambda^{-\frac{1}{2}} u)\rangle_{\mathcal{H}} \\ &\leq \langle Q_\lambda^{-\frac{1}{2}} u, dQ(Q_\lambda^{-\frac{1}{2}} u)\rangle_{\mathcal{H}} \\ &= d\langle u, QQ_\lambda^{-1} u\rangle_{\mathcal{H}} \\ &\leq d,\end{aligned}$$

and so

$$\|S\| \leq \mathcal{N}'_\infty(\lambda)\left(2\|Q_\lambda^{-\frac{1}{2}} M Q_\lambda^{-\frac{1}{2}}\| + 3d\|QQ_\lambda^{-1}\|\right) \leq 5d\mathcal{N}'_\infty(\lambda).$$

A lower bound can be obtained as

$$\begin{aligned}\|S\| &= \mathcal{N}'_\infty(\lambda)\left\|3dQ_\lambda^{-1}Q + 2Q_\lambda^{-\frac{1}{2}} M Q_\lambda^{-\frac{1}{2}}\right\| \\ &\geq \mathcal{N}'_\infty(\lambda)\left(\left\|3dQ_\lambda^{-1}Q\right\| - \left\|2Q_\lambda^{-\frac{1}{2}} M Q_\lambda^{-\frac{1}{2}}\right\|\right) \\ &\geq \mathcal{N}'_\infty(\lambda)\left(3d\frac{\|Q\|}{\|Q\|+\lambda} - 2d\right),\end{aligned}$$

so that when $\lambda \leq \rho\|Q\|$ we have that

$$\|S\| \geq d\mathcal{N}'_\infty(\lambda)\left(\frac{3}{1+\rho} - 2\right).$$

At last we can apply Lemma 12 to obtain that with probability at least $1-\delta$,

$$\lambda_{\max}\left(\frac{1}{n}Z_a\right) \leq \frac{2\beta'}{3n} + \sqrt{\frac{2\|S\|\beta'}{n}} \leq \frac{2\beta}{3n} + \sqrt{\frac{10d\mathcal{N}'_\infty(\lambda)\beta}{n}},$$

where

$$\beta' := \log\frac{2\operatorname{Tr} S}{\delta\|S\|} \leq \log\left(\frac{2}{\delta}\frac{5d\mathcal{N}'_\infty(\lambda)\operatorname{Tr} Q}{\lambda d\mathcal{N}'_\infty(\lambda)\left(\frac{3}{1+\rho} - 2\right)}\right) = \log\left(\frac{10\operatorname{Tr} Q}{\lambda\delta\left(\frac{3}{1+\rho} - 2\right)}\right) =: \beta$$

as required. □

7.4.3 Results on Hilbert space operators

Lemmas 14 and 15 were proven and used by [99].

Lemma 14 (Proposition 3 of [99]). *Let $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$ be three separable Hilbert spaces, with $Z : \mathcal{H}_1 \to \mathcal{H}_2$ a bounded linear operator and P a projection operator on $\mathcal{H}_1$ with* $\operatorname{range} P = \overline{\operatorname{range} Z^*}$. *Then for any bounded linear operator $F : \mathcal{H}_3 \to \mathcal{H}_1$ and any $\lambda > 0$,*

$$\|(I-P)F\| \le \sqrt{\lambda}\|(Z^*Z+\lambda I)^{-\frac{1}{2}}F\|.$$

Lemma 15 (Proposition 7 of [99]). *Let $\mathcal{H}$ be a separable Hilbert space, with A, B bounded self-adjoint positive linear operators on $\mathcal{H}$ and $A_\lambda = A + \lambda I$, $B_\lambda = B + \lambda I$. Then for any $\lambda > 0$,*

$$\|A_\lambda^{-\frac{1}{2}} B^{\frac{1}{2}}\| \le \|A_\lambda^{-\frac{1}{2}} B_\lambda^{\frac{1}{2}}\| \le (1-\gamma(\lambda))^{-\frac{1}{2}}$$

when

$$\gamma(\lambda) := \lambda_{\max}\left(B_\lambda^{-\frac{1}{2}}(B-A)B_\lambda^{-\frac{1}{2}}\right) < 1.$$

7.4.4 Distances between distributions in $\mathcal{P}$

Lemma 16 (Distribution distances from parameter distances). *Let $f_0, f \in \mathcal{F}$ correspond to distributions $p_0 = p_{f_0}, p = p_f \in \mathcal{P}$. Under Assumption (**H**), we have that for all $r \in [1, \infty]$:*

$$\begin{aligned}
\|p-p_0\|_{L^r(\Omega)} &\le 2\kappa e^{2\kappa\|f-f_0\|_{\mathcal{H}}} e^{2\kappa \min(\|f\|_{\mathcal{H}}, \|f_0\|_{\mathcal{H}})} \|f-f_0\|_{\mathcal{H}} \|q_0\|_{L^r(\Omega)} \\
\|p-p_0\|_{L^1(\Omega)} &\le 2\kappa e^{2\kappa\|f-f_0\|_{\mathcal{H}}} \|f-f_0\|_{\mathcal{H}} \\
\mathrm{KL}(f\|f_0) &\le c\kappa^2 \|f-f_0\|_{\mathcal{H}}^2 e^{\kappa\|f-f_0\|_{\mathcal{H}}} (1+\kappa\|f-f_0\|_{\mathcal{H}}) \\
\mathrm{KL}(f_0\|f) &\le c\kappa^2 \|f-f_0\|_{\mathcal{H}}^2 e^{\kappa\|f-f_0\|_{\mathcal{H}}} (1+\kappa\|f-f_0\|_{\mathcal{H}}) \\
h(f, f_0) &\le \kappa e^{\frac{1}{2}\|f-f_0\|_{\mathcal{H}}} \|f-f_0\|_{\mathcal{H}}
\end{aligned}$$

where c is a universal constant and h denotes the Hellinger distance $h(p, q) = \|\sqrt{p} - \sqrt{q}\|_{L^2(\Omega)}$.

Proof. First note that

$$\|f - f_0\|_\infty = \sup_{x\in\Omega} |f(x) - f_0(x)| = \sup_{x\in\Omega} |\langle f - f_0, k(x,\cdot)\rangle_{\mathcal{H}}| \leq \kappa \|f - f_0\|_{\mathcal{H}}.$$

Then, since each $f \in \mathcal{H}$ is bounded and measurable, $\mathcal{P}_\infty$ of Lemma A.1 of [117] is simply $\mathcal{P}$, and the result applies directly. □

Part III

Goodness-of-fit testing

Chapter 8

A Kernel Test of Goodness of Fit

This chapter is based on collaborative work, K. Chwialkowski, H. Strathmann, and A. Gretton. "A kernel test of goodness of fit". In: *International Conference for Machine Learning*. 2016.

We propose a non-parametric statistical test for goodness-of-fit: given a set of samples, the test determines how likely it is that these were generated from a target density function. The measure of goodness-of-fit is a divergence constructed via Stein's method using functions from a reproducing kernel Hilbert space. Our test statistic is based on an empirical estimate of this divergence, taking the form of a V-statistic in terms of the gradients of the log target density and of the kernel. We derive a statistical test, both for i.i.d. and non-i.i.d. samples, where we estimate the null distribution quantiles using a wild bootstrap procedure. We apply our test to quantifying convergence of approximate Markov chain Monte Carlo methods, statistical model criticism, and evaluating quality of fit in non-parametric score estimation.

Chapter outline

Recall that we are interested in the following question: given a set of samples from a distribution q, does q match some reference or target distribution p, which we assume to be only known up to the normalisation constant.

We begin in Section 8.1 with a high-level construction of the RKHS-based Stein discrepancy and associated statistical test. In Section 8.2, we provide additional details and prove the main results. Section 8.3 contains experimental illustrations on synthetic examples, statistical model criticism, bias-variance trade-offs in approximate MCMC, and convergence in non-parametric density estimation.

8.1 Test definition: statistic and threshold

We begin with a high-level construction of our divergence measure and the associated statistical test. While this section aims to outline the main ideas, we provide details and selected proofs in Section 8.2.

8.1.1 Stein operator in RKHS

Our goal is to write the maximum discrepancy between the target distribution p and observed sample distribution q in a *modified* RKHS, such that functions have zero expectation under p. Denote by $\mathcal{F}$ the RKHS of real-valued functions on $\mathbb{R}^d$ with reproducing kernel k, c.f. Section 2.1 and by $\mathcal{F}^d$ the product RKHS consisting of elements $f := (f_1, \ldots, f_d)$ with $f_i \in \mathcal{F}$, and with a standard inner product $\langle f, g \rangle_{\mathcal{F}^d} = \sum_{i=1}^d \langle f_i, g_i \rangle_{\mathcal{F}}$. We further assume that all measures considered in this paper are supported on an open set, equal to zero on the border, and strictly positive[1] (so logarithms are well defined). Similarly to Gorham and Mackey [52], Oates et al. [88], and Stein [118], we begin by defining a so-called *Stein operator* T_p acting on

[1] An example of such a space is the positive real line

$f \in \mathcal{F}^d$

$$(T_p f)(x) := \sum_{i=1}^{d} \left(\frac{\partial \log p(x)}{\partial x_i} f_i(x) + \frac{\partial f_i(x)}{\partial x_i} \right). \tag{8.1}$$

Suppose a random variable Z is distributed according to a measure[2] q and X is distributed according to the target measure p. As we will see, the operator can be expressed by defining a function that depends on gradient of the log-density and the kernel,

$$\xi_p(x,\cdot) := [\nabla \log p(x) k(x,\cdot) + \nabla k(x,\cdot)],$$

whose expected inner product with f gives exactly the expected value of the Stein operator,

$$\mathbb{E}_q T_p f(Z) = \langle f, \mathbb{E}_q \xi_p(Z) \rangle_{\mathcal{F}^d} = \sum_{i=1}^{d} \langle f_i, \mathbb{E}_q \xi_{p,i}(Z) \rangle_{\mathcal{F}},$$

where $\xi_{p,i}(x,\cdot)$ is the i-th component of $\xi_p(x,\cdot)$. For X from the target measure, we have $\mathbb{E}_p(T_p f)(X) = 0$, which can be seen using integration by parts, c.f. Lemma 17 in Section 8.2. We can now define a Stein discrepancy and express it in the RKHS,

$$\begin{aligned} S_p(Z) &:= \sup_{\|f\|<1} \mathbb{E}_q(T_p f)(Z) - \mathbb{E}_p(T_p f)(X) \\ &= \sup_{\|f\|<1} \mathbb{E}_q(T_p f)(Z) \\ &= \sup_{\|f\|<1} \langle f, \mathbb{E}_q \xi_p(Z) \rangle_{\mathcal{F}^d} \\ &= \| \mathbb{E}_q \xi_p(Z) \|_{\mathcal{F}^d}, \end{aligned} \tag{8.2}$$

This makes it clear why $\mathbb{E}_p(T_p f)(X) = 0$ is a desirable property: we can compute $S_p(Z)$ by computing $\| \mathbb{E}_q \xi_p(Z) \|$, without the need to access X in

[2] Throughout the chapter, all occurrences of Z, e.g. $Z', Z_i, Z_\heartsuit$, are understood to be distributed according to q.

the form of samples from p. To state our first result, we define

$$\begin{aligned} h_p(x,y) := {} & \nabla \log p(x)^\top \nabla \log p(y) k(x,y) \\ & + \nabla \log p(y)^\top \nabla_x k(x,y) \\ & + \nabla \log p(x)^\top \nabla_y k(x,y) \\ & + \langle \nabla_x k(x,\cdot), \nabla_y k(\cdot,y) \rangle_{\mathcal{F}^d}, \end{aligned}$$

where the last term can be written as a sum $\sum_{i=1}^d \frac{\partial k(x,y)}{\partial x_i \partial y_i}$. The following result gives a simple closed form expression for $\| \mathbb{E}_q \xi_p(Z) \|_{\mathcal{F}^d}$ in terms of h_p.

Theorem 5. *If $Eh_p(Z,Z) < \infty$, then $S_p^2(Z) = \| \mathbb{E}_q \xi_p(Z) \|^2_{\mathcal{F}^d} = \mathbb{E}_q h_p(Z,Z')$, where Z' is independent of Z with an identical distribution.*

The second main result states that the discrepancy $S_p(Z)$ can be used to distinguish two distributions.

Theorem 6. *Let q,p be probability measures and $Z \sim q$. If the kernel k is C_0-universal [30, Definition 4.1], $\mathbb{E}_q h_q(Z,Z) < \infty$, and $\mathbb{E}_q \left\| \nabla \left(\log \frac{p(Z)}{q(Z)} \right) \right\|^2 < \infty$, then $S_p(Z) = 0$ if and only if $p = q$.*

Before we proof these results in Section 8.2, we first proceed to construct an estimator for $S(Z)^2$, and outline its asymptotic properties.

8.1.2 Wild bootstrap testing

It is straightforward to estimate the squared Stein discrepancy $S(Z)^2$ from samples $\{Z_i\}_{i=1}^n$: a quadratic time estimator is a V-Statistic, and takes the form

$$V_n = \frac{1}{n^2} \sum_{i,j=1}^n h_p(Z_i, Z_j).$$

The asymptotic null distribution[3] of the normalised V-Statistic nV_n, however, has no computable closed form. Furthermore, care has to be taken

[3] The null distribution is the distribution of the estimator under the null hypothesis, here $p = q$, and an upper quantile of it is required for constructing a statistical test.

when the Z_i exhibit correlation structure, as the null distribution might significantly change, impacting test significance. This is particularly important when applied as an MCMC diagnosis tool. The wild bootstrap technique [45, 74, 107] addresses both problems. First, it allows us to estimate quantiles of the null distribution in order to compute test thresholds. Second, it accounts for correlation structure in the Z_i by mimicking it with an auxiliary random process: a simple Markov chain taking values in $\{-1,1\}$, starting from $W_{1,n} = 1$,

$$W_{t,n} = \mathbf{1}(U_t > a_n)W_{t-1,n} - \mathbf{1}(U_t < a_n)W_{t-1,n},$$

where the U_t are uniform $(0,1)$ i.i.d. random variables and a_n is the probability of $W_{t,n}$ changing sign (for i.i.d. data we set $a_n = 0.5$). This leads to a bootstrapped V-statistic

$$B_n = \frac{1}{n^2} \sum_{i,j=1}^{n} W_{i,n} W_{j,n} h_p(Z_i, Z_j).$$

Proposition 6 establishes that, under the null hypothesis, nB_n is a good approximation of nV_n, so it is possible to approximate quantiles of the null distribution by sampling from it. Under the alternative, V_n dominates B_n – resulting in almost sure rejection of the null hypothesis.

Statistical testing procedure

We propose the following test procedure for testing the null hypothesis that the Z_i are distributed according to the target distribution p.

1. Calculate the test statistic V_n.

2. Obtain wild bootstrap samples $\{B_n\}_{i=1}^{D}$ and estimate the $1 - \alpha$ empirical quantile of these samples.

3. If V_n exceeds the quantile, reject.

8.2 Proofs of the main results

We now prove the claims made in the previous section. We refer to Chwialkowski et al. [34] for further details, in particular in wild bootstrap testing.

8.2.1 Stein operator in RKHS

In order to proof Theorem 5, we establish Lemma 17, which shows that the expected value of the Stein operator is zero on the target measure.

Lemma 17. *If a random variable X is distributed according to p, under conditions on the kernel*

$$\begin{aligned} 0 &= \oint_{\partial\mathcal{X}} k(x,x')q(x)n(x)dS(x'), \\ 0 &= \oint_{\partial\mathcal{X}} \nabla_x k(x,x')^\top n(x')q(x')dS(x'), \end{aligned}$$

and then for all $f \in \mathcal{F}$, the expected value of T is zero, i.e. $\mathbb{E}_p(Tf)(X) = 0$.

Proof of Lemma 17. This result was proved on bounded domains $\mathcal{X} \subset \mathbb{R}^d$ by Oates et al. [88, Lemma 1], where $n(x)$ is the unit vector normal to the boundary at x, and $\oint_{\partial\mathcal{X}}$ is the surface integral over the boundary $\partial\mathcal{X}$. The case of unbounded domains was discussed by Oates et al. [88, Remark 2]. Here we provide an alternative, elementary proof for the latter case. First we show that the function $p \cdot f_i$ vanishes at infinity, by which we mean that for all dimensions j

$$\lim_{x_j \to \infty} p(x_1, \cdots, x_d) \cdot f_i(x_1, \cdots, x_d) = 0.$$

The density function p vanishes at infinity. The function f is bounded, which is implied by Cauchy-Schwarz inequality, $|f(x)| \leq \|f\| \sqrt{k(x,x)}$. This implies that the function $p \cdot f_i$ vanishes at infinity. We check that the expected value $\mathbb{E}_p(T_p)f(X)$ is zero. For all dimensions i,

$$\begin{aligned}
&\mathbb{E}_p(T_p)f(X)\\
&=\mathbb{E}_p\left(\frac{\partial \log p(X)}{\partial x_i}f_i(X)+\frac{\partial f_i(X)}{\partial x_i}\right)\\
&=\int_{R_d}\left[\frac{\partial \log p(x)}{\partial x_i}f_i(x)+\frac{\partial f_i(x)}{\partial x_i}\right]p(x)dx\\
&=\int_{R_d}\left[\frac{1}{p(x)}\frac{\partial p(x)}{\partial x_i}f(x)+\frac{\partial f(x)}{\partial x_i}\right]p(x)dx\\
&=\int_{R_d}\left[\frac{\partial p(x)}{\partial x_i}f_i(x)+\frac{\partial f_i(x)}{\partial x_i}p(x)\right]dx\\
&\overset{(a)}{=}\int_{R_{d-1}}\left(\lim_{R\to\infty}p(x)f_i(x)\Big|_{x_i=-R}^{x_i=R}\right)dx_1\cdots dx_{i-1}\cdots dx_{i+1}\cdots dx_d\\
&=\int_{R_{d-1}}0dx_1\cdots dx_{i-1}\cdots dx_{i+1}\cdots dx_d\\
&=0.
\end{aligned}$$

For the equation (a) we have used integration by parts, the fact that $p(x)f_i(x)$ vanishes at infinity, and the Fubini-Toneli theorem to show that we can do iterated integration. The sufficient condition for the Fubini-Toneli theorem is that $\mathbb{E}_q\langle f,\xi_p(Z)\rangle^2<\infty$. This is true since $\mathbb{E}_p\|\xi_p(X)\|^2\leq\mathbb{E}_p h_p(X,X)<\infty$. □

We now are ready to proof our result on the closed form Stein discrepancy.

Proof of Theorem 5. $\xi_p(x,\cdot)$ is an element of the reproducing kernel Hilbert space $\mathcal{F}^d$ – by Steinwart and Christmann [119, Lemma 4.34] $\nabla k(x,\cdot)\in\mathcal{F}$, and $\frac{\partial \log p(x)}{\partial x_i}$ is just a scalar. We first show that $h_p(x,y)=\langle\xi_p(x,\cdot),\xi_p(y,\cdot)\rangle$. Using notations

$$\begin{aligned}
\nabla_x k(x,\cdot)&=\left(\frac{\partial k(x,\cdot)}{\partial x_1},\cdots,\frac{\partial k(x,\cdot)}{\partial x_d}\right)\\
\nabla_y k(\cdot,y)&=\left(\frac{\partial k(\cdot,y)}{\partial y_1},\cdots,\frac{\partial k(\cdot,y)}{\partial y_d}\right),
\end{aligned}$$

we calculate

$$\begin{aligned}\langle \xi_p(x,\cdot), \xi_p(y,\cdot)\rangle &= \nabla \log p(x)^\top \nabla \log p(y) k(x,y) \\ &+ \nabla \log p(y)^\top \nabla_x k(x,y) \\ &+ \nabla \log p(x)^\top \nabla_y k(x,y) \\ &+ \langle \nabla_x k(x,\cdot), \nabla_y k(\cdot,y)\rangle_{\mathcal{F}^d}.\end{aligned}$$

Next we show that $\xi_p(x,\cdot)$ is Bochner integrable [see 119, Definition A.5.20],

$$\mathbb{E}_q \|\xi_p(Z)\|_{\mathcal{F}^d} \leq \sqrt{\mathbb{E}_q \|\xi_p(Z)\|^2_{\mathcal{F}^d}} = \sqrt{\mathbb{E}_q h_p(Z,Z)} < \infty.$$

This allows us to take the expectation inside the RKHS inner product. We next relate the expected value of the Stein operator to the inner product of f and the expected value of $\xi_q(Z)$,

$$\mathbb{E}_q T_p f(Z) = \langle f, \mathbb{E}_q \xi_p(Z)\rangle_{\mathcal{F}^d} = \sum_{i=1}^d \langle f_i, \mathbb{E}_q \xi_{p,i}(Z)\rangle_{\mathcal{F}}. \tag{8.3}$$

We check the claim for all dimensions,

$$\begin{aligned}&\langle f_i, \mathbb{E}_q \xi_{p,i}(Z)\rangle_{\mathcal{F}} \\ &= \left\langle f_i, \mathbb{E}_q \left[\frac{\partial \log p(Z)}{\partial x_i} k(Z,\cdot) + \frac{\partial k(Z,\cdot)}{\partial x_i}\right]\right\rangle_{\mathcal{F}} \\ &= \mathbb{E}_q \left\langle f_i, \frac{\partial \log p(Z)}{\partial x_i} k(Z,\cdot) + \frac{\partial k(Z,\cdot)}{\partial x_i}\right\rangle_{\mathcal{F}} \\ &= \mathbb{E}_q \left[\frac{\partial \log p(Z)}{\partial x_i} f_i(Z) + \frac{\partial f_i(Z,\cdot)}{\partial x_i}\right].\end{aligned}$$

The second equality follows from the fact that a linear operator $\langle f_i, \cdot\rangle_{\mathcal{F}}$ can be interchanged with the Bochner integral, and the fact that ξ_p is Bochner integrable. Using definition of $S(Z)$, Lemma 17, and (8.3), we have

$$\begin{aligned} S_p(Z) &:= \sup_{\|f\|<1} \mathbb{E}_q(T_p f)(Z) - \mathbb{E}_p(T_p f)(X) \\ &= \sup_{\|f\|<1} \mathbb{E}_q(T_p f)(Z) \\ &= \sup_{\|f\|<1} \langle f, \mathbb{E}_q \xi_p(Z)\rangle_{\mathcal{F}^d} \\ &= \| \mathbb{E}_q \xi_p(Z)\|_{\mathcal{F}^d}. \end{aligned}$$

We now calculate closed form expression for $S_p^2(Z)$,

$$\begin{aligned} S_p^2(Z) &= \langle \mathbb{E}_q \xi_p(Z), \mathbb{E}_q \xi_p(Z)\rangle_{\mathcal{F}^d} = \mathbb{E}_q \langle \xi_p(Z), \mathbb{E}_q \xi_p(Z)\rangle_{\mathcal{F}^d} \\ &= \mathbb{E}_q \langle \xi_p(Z), \xi_p(Z')\rangle_{\mathcal{F}^d} = \mathbb{E}_q h_p(Z, Z'), \end{aligned}$$

where Z' is an independent copy of Z. □

Next, we prove that the discrepancy S discriminates different probability measures.

Proof of Theorem 6. If $p = q$ then $S_p(Z)$ is 0 by Lemma 17. Suppose $p \neq q$, but $S_p(Z) = 0$. If $S_p(Z) = 0$ then, by Theorem 5, $\mathbb{E}_q \xi_p(Z) = 0$. In the following we substitute $\log p(Z) = \log q(Z) + [\log p(Z) - \log q(Z)]$,

$$\begin{aligned} &\mathbb{E}_q \xi_p(Z) \\ &= \mathbb{E}_q \left(\nabla \log p(Z) k(Z, \cdot) + \nabla k(Z, \cdot)\right) \\ &= \mathbb{E}_q \xi_q(Z) + \mathbb{E}_q \left(\nabla [\log p(Z) - \log q(Z)] k(Z, \cdot)\right) \\ &= \mathbb{E}_q \left(\nabla [\log p(Z) - \log q(Z)] k(Z, \cdot)\right) \end{aligned}$$

We have used Theorem 5 and Lemma 17 to see that $\mathbb{E}_q \xi_q(Z) = 0$, since $\| \mathbb{E}_q \xi_q(Z)\|^2 = S_q^2(Z) = 0$.

We recognise that the expected value of $\nabla(\log p(Z) - \log q(Z)) k(Z, \cdot)$

is the mean embedding of a function $g(y) = \nabla\left(\log\frac{p(y)}{q(y)}\right)$ with respect to the measure q. By the assumptions the function g is square integrable; therefore, since the kernel k is C_o-universal, by Carmeli et al. [30, Theorem 4.2 b] its embedding is zero if and only if $g = 0$. This implies that

$$\nabla \log \frac{p(y)}{q(y)} = (0, \cdots, 0).$$

A constant vector field of derivatives can only be generated by a constant function, so $\log\frac{p(y)}{q(y)} = C$, for some C, which implies that $p(y) = e^C q(y)$. Since p and q both integrate to one, $C = 0$ and thus $p = q$, which is a contradiction. □

8.2.2 Wild bootstrap testing

The following result justifies our proposed test in Section 8.1.2.

Proposition 5. If h is Lipschitz continuous and $\mathbb{E}_q h_p(Z, Z) < \infty$ then, under the null hypothesis, nV_n converges weakly to some distribution.

The proof, which is a simple verification of the relevant assumptions, can be found in Chwialkowski et al. [34]. Although a formula for a limit distribution of V_n can be derived explicitly [73, Theorem 2.1], we do not provide it here. To our knowledge there are no methods of obtaining quantiles of a limit of V_n in closed form. The common solution is to estimate quantiles by a re-sampling method, as described in Section 8.1. The validity of this re-sampling method is guaranteed by the following proposition (which follows from Theorem 2.1 of Leucht and a modification of the Lemma 5 of Chwialkowski et al. [36]), see Chwialkowski et al. [34] for a proof.

Proposition 6. Let $f(Z_{1,n}, \cdots, Z_{t,n}) = \sup_x |P(nB_n > x | Z_{1,n}, \cdots, Z_{t,n}) - P(nV_n > x)|$ be a difference between quantiles. If h is Lipschitz continuous and $\mathbb{E}_q h_p(Z, Z)^2 < \infty$ then, under the null hypothesis, $f(X_{1,n}, \cdots, X_{t,n})$

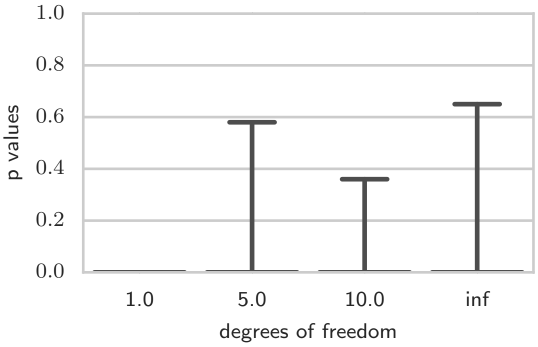


Figure 8.1: Large autocorrelation, unsuitable bootstrap. The parameter a_n is too large and the bootstrapped V-statistics B_n are too low on average. Therefore, it is very likely that $V_n > B_n$ and the test is too conservative.

converges to zero in probability; under the alternative hypothesis, B_n converges to zero, while V_n converges to a positive constant.

As a consequence, if the null hypothesis is true, we can approximate any quantile; while under the alternative hypothesis, all quantiles of B_n collapse to zero while $P(V_n > 0) \to 1$. For the specific case of testing MCMC convergence, we point to our discussion in Chwialkowski et al. [34, Appendix].

8.3 Experiments

We provide a number of experimental applications for our test. We begin with a simple check to establish correct test calibration on non-i.i.d. data, followed by a demonstration of statistical model criticism for Gaussian process (GP) regression. We then apply the proposed test to quantify bias-variance trade-offs in MCMC, and demonstrate how to use the test to verify whether MCMC samples are drawn from the desired stationary distribution. In the final experiment, we move away from the MCMC setting, and use the test to evaluate the convergence of a non-parametric density estimator.

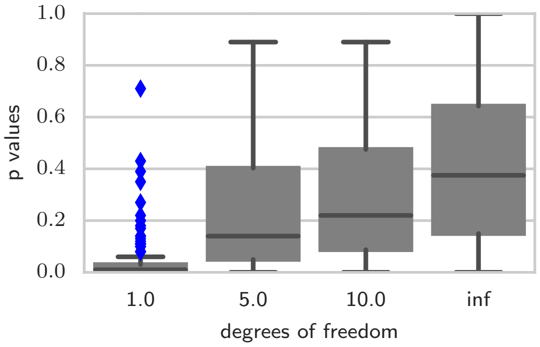


Figure 8.2: Large auto-covariance, suitable bootstrap. The parameter $a_n$is chosen suitably, but due to a large auto-covariance within the samples, the power of the test is small (effective sample size is small).

8.3.1 Student's t vs. Normal

In our first experiment, performed by co-author Kacper Chwialkowski, we modify Experiment 4.1 from Gorham and Mackey [52]. The null hypothesis is that the observed samples come from a standard normal distribution. We study the power of the test against samples from a Student's t distribution. We expect to observe low p-values when testing against a Student's t distribution with few degrees of freedom. We consider 1, 5, 10 or ∞ degrees of freedom, where ∞ is equivalent to sampling from a standard normal distribution. For a fixed number of degrees of freedom we draw 1400 samples and calculate the p-value. This procedure is repeated 100 times, and the bar plots of p-values are shown in Figure 8.1, Figure 8.2, and Figure 8.3.

Our twist on the original Experiment 4.1 by Gorham and Mackey is that the draws from the Student's t distribution exhibit temporal correlation. We generate samples using a Metropolis–Hastings algorithm, with a Gaussian random walk with variance $\frac{1}{2}$. We emphasise the need for an appropriate choice of the wild bootstrap process parameter a_n. In Figure 8.1 we plot p-values for a_n being set to 0.5. Such a high value of a_n is suitable for i.i.d. observations, but results in p-values that are too conservative for temporally correlated observations. In Figure 8.2, we set $a_n = 0.02$, which gives a well calibrated distribution of the p-values under the null hypothe-

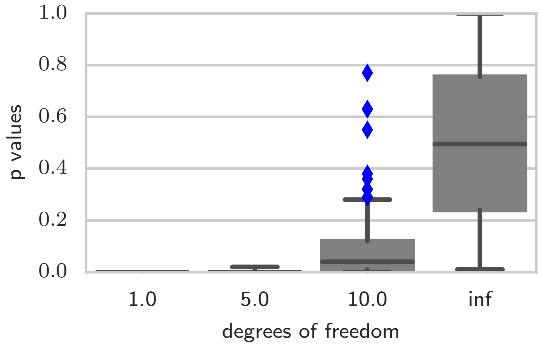


Figure 8.3: Thinned sample, suitable bootstrap. Most of the autocorrelation within the sample is cancelled by thinning. To guarantee that the remaining autocorrelation is handled properly, the wild bootstrap flip probability is set at 0.1.

sis, however the test power is reduced. Indeed, p-values for five degrees of freedom are already large. The solution that we recommend is a mixture of thinning and adjusting a_n, as presented in the Figure 8.3. We thin the observations by a factor of 20 and set $a_n = 0.1$, thus preserving both good statistical power and correct calibration of p-values under the null hypothesis. In a general, we recommend to thin a chain so that $\mathrm{Cor}(X_t, X_{t-1}) < 0.5$, set $a_n = 0.1/k$, and run test with at least $\max(500k, d100)$ data points, where $k < 10$.

8.3.2 Comparing to a parametric test in increasing dimensions

In this experiment, performed by co-author Kacper Chwialkowski, we compare with the test proposed by Baringhaus and Henze [13], which is essentially an MMD test for normality, i.e. the null hypothesis is that Z is a d-dimensional standard normal random variable. We set the sample size to $n = 500, 1000$ and $a_n = 0.5$, generate

$$Z \sim \mathcal{N}(0, I_d) \qquad Y \sim U[0,1],$$

	d	**2**	**5**	**10**	**15**	**20**	**25**
B&H	$n = 500$	1	1	**1**	**0.86**	**0.29**	**0.24**
Stein		1	1	0.86	0.39	0.05	0.05
B&H	$n = 1000$	1	1	1	**1**	**0.87**	**0.62**
Stein		1	1	1	0.77	0.25	0.05

Table 8.1: Test power vs. sample size for the test by Baringhaus and Henze [13] (B&H) and our Stein based test.

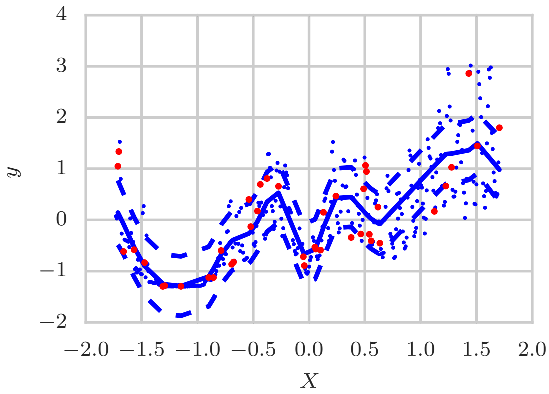


Figure 8.4: Fitted GP and data used to fit (blue) and to apply test (red).

and modify $Z_0 \leftarrow Z_0 + Y$. Table 8.1 shows the power as a function of the sample size. We observe that for higher dimensions, and where the expectation of the kernel exists in closed form, an MMD-type test like [13] is a better choice.

8.3.3 Statistical model criticism on Gaussian processes

We next apply our test to the problem of statistical model criticism for GP regression. Our presentation and approach are similar to the non i.i.d. case in Section 6 of Lloyd and Ghahramani [79]. We use the `solar` dataset, consisting of a $d = 1$ regression problem with $N = 402$ pairs (X, y). We fit $N_{\text{train}} = 361$ data using a GP with an exponentiated quadratic kernel and a Gaussian noise model, and perform standard maximum likelihood II on the hyper-parameters (length-scale, overall scale, noise-variance). We then apply our test to the remaining $N_{\text{test}} = 41$ data. The test attempts to falsify the null hypothesis that the `solar` dataset was generated from the plug-in predictive distribution (conditioned on training data and predicted position) of the GP. Lloyd and Ghahramani refer to this set-up as non-i.i.d., since the predictive distribution is a different univariate Gaussian for every predicted point. Our particular $N_{\text{train}}, N_{\text{test}}$ were chosen to make sure the GP fit has stabilised, i.e. adding more data did not cause further model refinement.

Figure 8.4 shows training and testing data, and the fitted GP. Clearly, the Gaussian noise model is a poor fit for this particular dataset, e.g. around $X = -1$. Figure 8.5 shows the distribution over $D = 10000$ bootstrapped V-statistics B_n with $n = N_{\text{test}}$. The test statistic lies in an upper quantile of the bootstrapped null distribution, correctly indicating that it is unlikely the test points were generated by the fitted GP model, even for the low number of test data observed, $n = 41$.

In a second experiment, we compare against Lloyd and Ghahramani: we compute the MMD statistic, c.f. Section 2.1, between test data $(X_{\text{test}}, y_{\text{test}})$ and $(X_{\text{test}}, y_{\text{rep}})$, where y_{rep} are samples from the fitted GP. We draw 10000 samples from the null distribution by repeatedly sampling new $\tilde{y}_{\text{rep}}$ from the GP plug-in predictive posterior, and comparing $(X_{\text{test}}, \tilde{y}_{\text{rep}})$ to $(X_{\text{test}}, y_{\text{rep}})$. When averaged over 100 repetitions of randomly partitioned (X, y) for training and testing, our goodness of fit test produces a p-value that is statistically not significantly different from the MMD method ($p \approx 0.1$, note that this result is subject to $N_{\text{train}}, N_{\text{test}}$). We emphasise, however, that Lloyd and Ghahramani's test requires to sample from the fitted model (here 10000 null samples were required in order to achieve stable p-values). Our test *does not* sample from the GP at all and completely sidesteps this more costly approach.

8.3.4 Bias quantification in approximate MCMC

We now illustrate how to quantify bias-variance trade-offs in an approximate MCMC algorithm – austerity MCMC [69]. For the purpose of illustration we use a simple generative model from Gorham and Mackey [52]

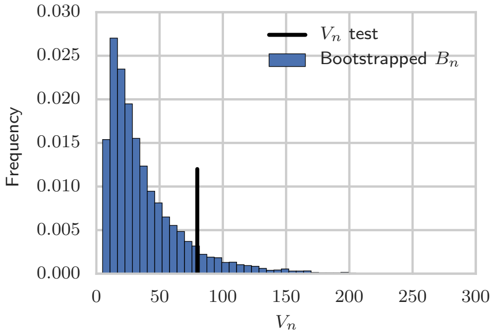


Figure 8.5: Bootstrapped B_n distribution for the GP experiment, with the test statistic V_n marked.

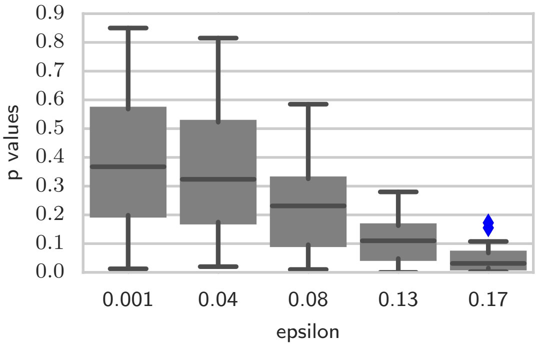


Figure 8.6: Distribution of p-values as a function of ϵ for austerity MCMC.

and Welling and Teh [128],

$$\theta_1 \sim \mathcal{N}(0,10); \theta_2 \sim \mathcal{N}(0,1)$$
$$X_i \sim \frac{1}{2}\mathcal{N}(\theta_1,4) + \frac{1}{2}\mathcal{N}(\theta_2+\theta_1,4).$$

Austerity MCMC is a Monte Carlo procedure designed to reduce the number of likelihood evaluation in the acceptance step of the Metropolis-Hastings algorithm. The crux of method is to look at only a subset of the data, and make an acceptance/rejection decision based on this subset. The probability of making a wrong decision is proportional to a parameter $\epsilon \in [0,1]$. This parameter influences the time complexity of austerity MCMC: when ϵ is larger, i.e., when there is a greater tolerance for error, the expected computational cost is lower. We simulate $\{X_i\}_{1\leq i\leq 400}$ points

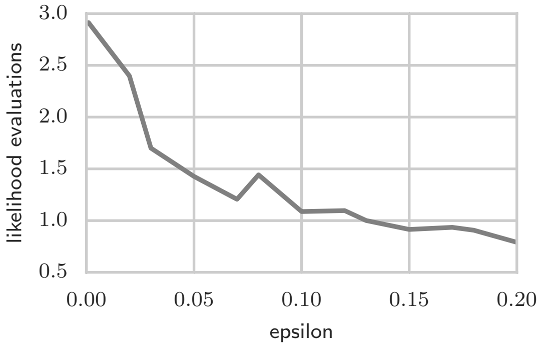


Figure 8.7: Average number of likelihood evaluations a function of ϵ for austerity MCMC (the y-axis is in millions of evaluations).

from the model with $\theta_1 = 0$ and $\theta_2 = 1$. In our experiment, there are two modes in the posterior distribution: one at $(0,1)$ and the other at $(1,-1)$. We run the algorithm with ϵ varying over the range $[0.001, 0.2]$. For each ϵ we calculate an individual thinning factor, such that correlation between consecutive samples from the chains is smaller than 0.5 (greater ϵ generally requires more thinning). For each ϵ we test the hypothesis that $\{\theta_i\}_{1 \le i \le 500}$ is drawn from the true stationary posterior, using our goodness of fit test. We generate 100 p-values for each ϵ, as shown in Figure 8.6. A good approximation of the true stationary distribution is obtained at $\epsilon = 0.4$, which is still parsimonious in terms of likelihood evaluations, as shown in Figure 8.7.

8.3.5 Convergence in non-parametric density estimation

In our final experiment, which relates back to the density models described in Part I, we apply our goodness of fit test to measuring quality-of-fit in non-parametric density estimation. We evaluate two density models: the original infinite dimensional exponential family [117], described in Section 2.1 with the estimator outlined in Chapter 6, and a the random Fourier features approximation from Section 4.2.2. Our implementation of the model assumes the log density to take the form $f(x)$, where f lies in an RKHS induced by a Gaussian kernel with bandwidth 1. We fit the model using N observations drawn from a standard Gaussian, and per-

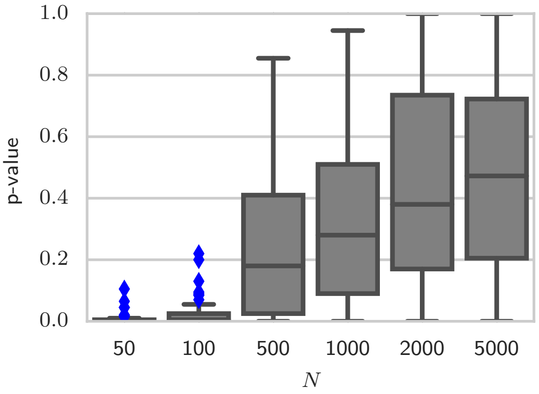


Figure 8.8: Density estimation: p-values for an increasing number of data N for the non-parametric model. Fixed $n = 500$.

form our quadratic time test on a separate evaluation dataset of fixed size $n = 500$. Our goal is to identify N sufficiently large that the goodness of fit test does not reject the null hypothesis (i.e. the model has learned the density sufficiently well, bearing in mind that it is guaranteed to converge for sufficiently large N). Figure 8.8 shows how the distribution of p-values evolves as a function of N; this distribution is uniform for $N = 5000$, but at $N = 500$, the null hypothesis would very rarely be rejected.

We next consider the random Fourier feature approximation to this model, where the log pdf f is approximated using a finite dictionary of random Fourier features [89]. The natural question when using this approximation is: 'How many random features are needed?' Using the same test set size $n = 500$ as above, and a large number of samples, $N = 5 \cdot 10^4$, Figure 8.9 shows the distributions of p-values for an increasing number of random features m. From $m = 50$, the null hypothesis would rarely be rejected. Note, however, that the p-values do *not* have a uniform distribution, even for a large number of random features. This subtle effect is caused by over-smoothing due to the (non-decreasing L_2-) regularisation approach taken in Section 4.2.2, which would not otherwise have been detected.

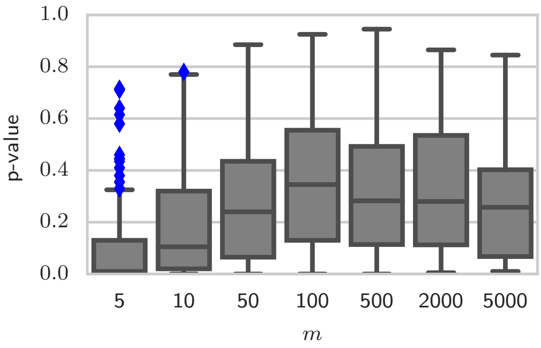


Figure 8.9: Approximate density estimation: p-values for an increasing number of random features m. Fixed $n = 500$.

Part IV

Conclusions

Adaptive Monte Carlo proposals

We discuss a number of points that are shared by adaptive Metropolis-Hastings in Chapter 3, gradient-free HMC in Chapter 4, and adaptive SMC in Chapter 5.

We have seen that building surrogate models for the underlying density can be used to improve sampling efficiency of MCMC and SMC algorithms. In particular, kernel methods offer a flexible and efficient way to model covariance and gradient structure that can be used in those surrogate models. We developed three algorithm classes that attempt to model the density using samples from the Markov chain history or from the current set of weighted SMC particles.

The most interesting use-case is when the underlying density is intractable (apart from having a non-trivial or non-linear support), that is it can only be estimated point-wise without bias, but neither the true log-pdf nor its gradients are available in closed form. In our examples in Section 3.3.1, Section 4.4.3, Section 5.3.3, and Section 5.3.4, intractability arose from integrating out latent variables in order to turn complicated joint distributions into marginal likelihoods with simpler structure. Our work here closely aligns with the literature on pseudo-marginal MCMC [6, 44], and nested importance sampling techniques [33, 125]. Compared to the random walk methods usually used in these settings, our kernel adaptive proposals lead to faster convergence in many cases.

MCMC, SMC, and diminishing adaptation

In Chapter 5, we have seen that the adaptive proposal mechanisms developed for MCMC in Chapter 3 and Chapter 4 can be embedded into the SMC framework in a straight-forward manner, with minor adjustments to account for weighted sample sets, e.g. the weighted KMC finite gradient estimator in (5.2).

MCMC algorithms with adaptive proposal distributions are not generally guaranteed to give consistent estimators. That is, continuous re-fitting or updating of the surrogate model as the Markov chain grows is not guaranteed to result in a Markov chain that converges to the correct stationary distribution, if any. For the adaptive Metropolis proposal by Haario et al. [60], it was shown that under certain assumptions on the target density, the proposal mechanism stabilises for continuous adaptation [4]. This is not necessarily true for our proposed KAMH algorithm, and even less so for the gradient-based KHMC. These samplers require a vanishing adaptation schedule, in order to ensure convergence to the correct target [97], c.f. Section 3.2.2, Section 4.3; or even a complete stop as in Theorem 1. Since the presented algorithms fall back to random walk behaviour in practice, stopping adaptation is not compromising efficiency compared to random walk schemes, c.f. Theorem 1.

The need to choose a vanishing adaptation schedule creates a difficult to tune exploration-exploitation trade-off with limited principled guidance. In Chapter 5, we saw that SMC is a more natural framework for employing RKHS-based representations. SMC proposals can continuously be adapted and the choice of an adaptation schedule is thus entirely circumvented.

Somewhat ironically, however, the efficiency gains in our experiments in Section 5.3 are less significant than in the MCMC case, c.f. Section 3.3, Section 4.4. An explanation for the case of kernel gradient importance sampling (KGRIS) in Section 5.2.2 might be the fact that PMC proposal is generated using only a *single* gradient step, following [101, GRIS], whereas the MCMC proposal along the kernel induced Hamiltonian flow in Chapter 4 involved *multiple* gradient steps. At the date of writing, however, there exists only initial work addressing the use of Hamiltonian dynamics in importance sampling [85].

Sampling efficiency

While adaptive MCMC transition kernel proposals can increase statistical efficiency of the underlying sampling scheme, they impose additional computational costs.

For example, drawing from the kernel covariance based KAMH proposal in Proposition 2 in Chapter 3 costs $\mathcal{O}(n^2d + d^3)$ in every step, where n is the number of data used and d is their dimensionality. The KMC lite estimator in Proposition 3 in Chapter 4 costs $\mathcal{O}(n^3 + dn^2)$ when fitting, and $\mathcal{O}(dn)$ for every leap-frog step. The costs of the SMC rejuvenation moves in Chapter 5 are similar.

Somewhat surprisingly, however, these relatively large costs do not severely impact the runtime efficiency in practice. The reason is that in the context of intractable likelihoods, the computational cost of fitting a density model is typically dominated by the larger cost of evaluating the model likelihood. In our real-world experiments on GP classification and a stochastic volatility model in Section 5.3.3, Section 3.3.1, Section 4.4.3 and Section 5.3.4, a profiler reveals that only roughly less than 5% of the overall wall-clock time is spent in generating the proposals. This effect increases with dataset size and model complexity, as evaluating likelihood gets more costly. On the other hand, in the case where we need not resort to pseudo-marginal or SMC^2 type samplers, the application of kernel based proposals might not result in a better statistical efficiency per time.

High dimensionality

In chapter Chapter 4, we have seen that the developed gradient estimator scale up from dozens to a hundred dimensions on laptop computers, depending on smoothness properties of the target. Eventually, however, the number of data required to sufficiently estimate non-linear covariance and gradients quickly becomes infeasible, c.f. Section 4.4.1 and the experiments therein. High dimensional sampling problems typically arise in non-parametric models, e.g. the Gaussian processe experiment in Section 3.3.1, or the stochastic volatility model in Section 5.3.4, where each data point increases the number of parameters. In the intractable likelihood framework that we consider here, however, the marginal posterior over hyper-parameters typically is independent of such latent variables – and therefore usually of moderate dimension. Random walk methods, which are the default choice for intractable likelihoods, scale badly in high dimensions themselves [86]. Our method is an improvement in the intermediate case: closed form gradients are not available, but the dimensionality of the problem allows to estimate the target geometry just accurately enough to improve mixing.

It is an active area of research to further scale up these techniques by exploiting structure in the target density, for example by assuming a pairwise structure [121, 135].

Kernel surrogate as a posterior approximation

A common reviewer comment [101, 120] was that the kernel approximation of the target density could be considered itself as an output of our algorithms, representing the posterior directly. There are a number of problems with this approach:

- Our density models do not need to be perfect to generate useful proposals, therefore allowing us to exploit posterior structure much earlier (even with non-perfect model fit) during sampling, still resulting in a correct (and efficient) sampler, c.f. Theorem 1.

- Approximating integrals of test functions with respect to the posterior using the kernel model is not possible in closed form. For example, take the KMC lite model (4.3) (with a Gaussian kernel), whose density is the exponential of a sum of Gaussian kernels centred at the data X_i. Computing an integral as simple as the posterior mean,

  $$Z^{-1}\int x \exp\left(\sum_i \alpha_i \exp(-\|X_i - x\|^2)\right) \mathrm{d}x,$$

  already is intractable, even if the evidence Z was known.

- It is not possible to sample from the kernel density model directly. One could imagine running a second sampler (e.g. HMC) targeting the surrogate model, an approach taken as a sub-module by Zhang et al. [137]. The generated samples, however, are not guaranteed to consistently estimate posterior expectations. Therefore, without a clear notion of consistency and accuracy of the density model, it is unclear to what extend bias is introduced.

Improving HMC runtime for large datasets

We have seen in the approximate Bayesian computation example in Section 4.4.4 that one of the benefits of using a surrogate gradient for HMC is the reduced number of ABC-likelihood simulations. Compared to the stochastic finite difference method SPAS by Meeds et al. [82], which needs to simulate from the likelihood in every leap-frog step, KHMC only does this *once* per iteration.

A similar argument can be made for posteriors with a large number of likelihood evaluations, whose gradient *is available*. Via replacing the Hamiltonian flow (4.1) with a surrogate (kernel) based flow, every leap-frog step only requires evaluation of the surrogate gradient, c.f. Section 4.1. Consequently, if the number of likelihood evaluations is large, a large runtime saving can be achieved compared to running classical HMC. Zhang et al. [136] combine this idea with surrogate models based on random basis functions, similar to KMC finite (4.5). Ultimately, efficiency of such an approach strongly depends on smoothness of the target density – as for KHMC, the acceptance rate drops with the accuracy of the surrogate model. This once more motivates the need for theoretical guarantees of such surrogate models, addressed in Chapter 6.

Efficient and principled score estimation

We have developed and studied a theoretically founded Nyström approximation to the infinite exponential family model in Section 2.1. Here we comment on relevance, alternative proof techniques, the relationship to the approximations in Chapter 4, and avenues for future research.

Relevance

We first emphasise that simply applying the Nyström technique to an algorithm is a much smaller contribution than proving that it has the same generalisation ability as the exact solution with $m = o(n)$, i.e. proven cost savings without loss of generalisation. Establishing this for ridge regression took 15 years, from the works of Williams and Seeger [130] to those of Rudi et al. [99].

Re-using generalisation bounds from regression?

On first sight, one might challenge the need for the long and technical proof in Chapter 6. In fact, the infinite exponential family estimator (2.11) by Sriperumbudur et al. [117] has a very similar form to standard kernel ridge regression: a single linear solve in (2.13) where the inverted matrix is contains kernel entries between input data (here, the system matrix $G \in \mathbb{R}^{nd \times nd}$ contains kernel partial derivatives $G_{(a,i),(b,j)} = \partial_i \partial_{j+d} k(X_a, X_b,))$. Therefore, it seems tempting to re-cast the problem as a regression problem, and to directly apply the Nyström generalisation bounds by Rudi et al. [99]. Unfortunately, this strategy does not work for two reasons.

To understand the first problem, we rewrite the score matching loss so that the Nyström regression results by Rudi et al. apply. Define

$$q_{\lambda,n} := \sum_{a=1}^{n} \sum_{i=1}^{d} \beta_{(a,i)} \partial_i k(X_a, \cdot)$$

such that

$$f_{\lambda,n} = q_{\lambda,n} - \frac{\hat{\xi}}{\lambda},$$

where $f_{\lambda,n}$ is the full solution in (2.11). The Nyström approximation then becomes

$$\begin{aligned} q^m_{\lambda,n} &= \operatorname{argmin}_{q\in\mathcal{H}_Y} \hat{J}_\lambda\left(q - \frac{\hat{\xi}}{\lambda}\right) \\ &= \operatorname{argmin}_{q\in\mathcal{H}_Y}\left(\|Z_X q - \hat{y}_n\|^2_{\mathcal{H}} + \lambda\,\|q\|^2_{\mathcal{H}}\right) \\ &= \frac{1}{\lambda} g_Y(\hat{C}_\lambda)\hat{C}\hat{\xi}, \end{aligned}$$

where Z_X is the 'derivative evaluation' operator defined in (7.3), $g_Y(\cdot)$ was defined in (7.7) with properties outlined in Lemma 2, and the connection to ridge regression is via the

$$\hat{y}_n := \frac{1}{\lambda} Z_X \hat{\xi},$$

which take the role of 'regression labels'[4]. We now have a ridge regression problem, and therefore can follow Rudi et al. [99, Theorem 2]. Start with

$$f^m_{\lambda,n} - f_{\lambda,n} = \frac{1}{\lambda}\underbrace{\left[g_Y(\hat{C}_\lambda)\hat{C}_\lambda - I\right]}_{(*)}\hat{C}^{-1}_\lambda \hat{C}\hat{\xi}. \tag{8.4}$$

Bounding the norm of the term $(*)$ as in Rudi et al. [99] quickly leads to a norm

$$\|C^{1/2}_\lambda(I - VV^*)\|,$$

which drops as $\lambda^{1/2}$ with high probability ([99, Lemma 6], though also note the second problem below). This means the overall order of $f^m_{\lambda,n} - f_{\lambda,n}$ in (8.4), due to the $\hat{C}^{-1}_\lambda$, is $\lambda^{-1/2}$, which *increases* as $\lambda \to 0$, and thus cannot lead to a vanishing error bound.

The second problem comes from the definition of the covariance operator C in (7.1) (and related operators), defined in terms of a sum of derivatives. Due to the correlation among dimensions in this operator, our concentration result on sums of correlated random operators in Lemma 13 is

[4] In contrast to regression, the 'labels' $\hat{y}_n$ here depend both on the regularisation parameter λ and the input data X_a via the definition of ξ in (2.12).

far more complex than the equivalent of Rudi et al. [99, Proposition 8]. Even if the regression approach worked, we would not have be able to use their convergence lemmas.

Relationship of Nyström and 'lite' kernel exponential families

We show that the lite kernel exponential family in Proposition 3 in Chapter 4 is a special case of the Nyström framework, as already discussed in Section 6.1.1. The lite estimator obtains a solution in $\mathcal{H}'_Y = \operatorname{span}\{k(y,\cdot)\}_{y\in Y}$, where we assumed that $Y = X$, $k(x,y) = \exp\left(-\sigma^{-1}\|x-y\|^2\right)$, and q_0 is uniform. Recall the estimator in Proposition 3,

$$\alpha = -\frac{\sigma}{2}(A+\lambda I)^{-1}b \tag{8.5}$$

$$A = \sum_{i=1}^{d} -[D_{x_i}K - KD_{x_i}]^2 \qquad b = \sum_{i=1}^{d}\left(\frac{2}{\sigma}(Ks_i + D_{s_i}K\mathbf{1} - 2D_{x_i}Kx_i) - K\mathbf{1}\right)$$

where $x_i = \begin{bmatrix} X_{1i} & \dots & X_{ni}\end{bmatrix}^\mathsf{T}$, $s_i = x_i \odot x_i$ with $\odot$ the element-wise product, $D_x = \operatorname{diag}(x)$, and $K \in \mathbb{R}^{m\times m}$ has entries $K_{aa'} = k(X_a, X_{a'})$.

Lemma 4 allows us to optimize over $\mathcal{H}'_Y$; we need not restrict ourselves to $Y = X$, uniform q_0, or a Gaussian kernel. Here $y_a = k(Y_a,\cdot)$, and we obtain

$$\beta'_Y = -\left(\frac{1}{n}(B'_{XY})^\mathsf{T}B'_{XY} + \lambda G'_{YY}\right)^\dagger h'_Y.$$

Using that for the Gaussian kernel k

$$\partial_i k(x,y) = -\frac{2}{\sigma}(x_i - y_i)k(x,y) \qquad \partial^2_{i+d}k(x,y) = \frac{2}{\sigma}\left[\frac{2}{\sigma}(x_i-y_i)^2 - 1\right]k(x,y),$$

we can obtain with some algebra similar to the proof of Proposition 3 that when $Y = X$ and q_0 is uniform,

$$h'_X = \frac{2}{n\sigma}b \qquad (B'_{XX})^\mathsf{T}B'_{XX} = \frac{4}{\sigma^2}A \qquad G'_{XX} = K.$$

Thus

$$\beta'_X = -\left(\frac{4}{n\sigma^2}A + \lambda K\right)^{\dagger}\frac{2}{n\sigma}b = -\frac{\sigma}{2}\left(A + \frac{1}{4}n\sigma^2\lambda K\right)^{\dagger} b. \tag{8.6}$$

(8.6) resembles (8.5) and Proposition 3, except that now, we regularise A with $\frac{1}{4}n\sigma^2\lambda K$ rather than λI, i.e. using the RKHS norm rather than the L_2 norm respectively. As mentioned in the experiments in Section 6.3, adding a small L_2 term improves numerical stability further.

Future work: establishing theory for 'lite Nyström'

The **lite** estimator from Chapter 4 performs strong empirically, especially when considering runtime efficiency in growing dimensions – its computational costs do not grow cubically in d as opposed to **nyström** from Chapter 6. Therefore, it is desirable to establish theoretical guarantees for this estimator as well, i.e. to extend the theoretical framework developed in this section to minimising the Fisher loss over $\mathcal{H}'_Y = \text{span}\{k(x,\cdot)\}_{x\in X}$, c.f. Section 6.1.1. This in particular concerns the 'approximation error' term in the decomposition (6.3),

$$\|f^m_{\lambda,n} - f_0\|_{\mathcal{H}} \leq \|f^m_{\lambda,n} - f^m_{\lambda}\|_{\mathcal{H}} + \underbrace{\|f^m_{\lambda} - f_0\|_{\mathcal{H}}}_{\text{Approximation error}} .$$

The space over which the lite approximation optimises is a subspace of the full estimator's RKHS, and establishing theory therefore will involve quantifying how fast $\mathcal{H}'_Y$ covers f_0 as opposed to our used $\mathcal{H}_Y$.

Rudi et al. [98] recently developed theoretical guarantees for the case of using an explicit random Fourier basis for regression, and their used techniques might serve as inspiration for our context.

Goodness-of-fit testing

The kernel Stein discrepancy (KSD) from Chapter 8 lead up to a number of interesting follow-up works, of which we give three examples.

Stein variational gradient descent

Liu and Wang [78] proposed an approximate Bayesian inference technique based on the KSD. Their method moves particles to match a desired posterior distribution. This is achieved via a form of functional gradient descent that minimises the KL divergence between the particle approximation and the true posterior iteratively. The method is justified by an established connection between the derivative of the KL divergence with the Stein operator [78, Theorem 1], and this allows for computation of the functional gradient via only accessing the score function of the true density [78, Algorithm 1].

Linear time goodness-of-fit testing

Recently, Jitkrittum et al. [66] developed a linear time version of our goodness-of-fit test. Their work is based on the idea of 'fast analytic function representation' of probability measures, as advocated by Chwialkowski et al. [35]. The main idea comes from the fact that the KSD in (8.2) is the RKHS norm of a *witness function*. When a real analytic kernel is used, it suffices to evaluate this witness function at only finitely many points (drawn from a density) in order to decide whether it is likely to be zero everywhere.

Jitkrittum et al. exploit this fact and develop a linear time goodness-of-fit test that has higher empirical test power than a naive linear time version, as for example suggested by Liu et al. [77, Equation 17].

Recent work on KSD for score estimation

Li and Turner [75] used the the KSD to construct a score estimator similar to the kernel exponential family 'lite' model in Chapter 4, and as such their approach connects Chapter 4 and Chapter 8 of this work. Recall Lemma 17 that states that the expected value of the Stein operator T_p in (8.1) is zero on the target measure, i.e. for $X \sim p$ and all test functions $f \in \mathcal{F}$, we have that

$$\mathbb{E}_p(T_p f)(X) = \mathbb{E}_p \left[\sum_{i=1}^{d} \partial_i \log p(X) f_i(X) + \partial_i f_i(X) \right] = 0.$$

Li and Turner propose to 'invert' a Monte Carlo version of this identity, i.e. to solve it for the gradients $\partial_i \log p(X_a)$ via ridge regression. The result is a matrix collecting all learned gradients $\partial_i \log p(X_a)$ at all training locations $\{X_a\}_{a=1}^n$.

Their approach is non-parametric in the sense that it does not generally assume a functional form of the gradient model, however, this comes at the cost of not being able to evaluate the learned gradient outside the training data [75, Section 3.4]. This renders the approach unusable for example for our kernel HMC method in Chapter 4. Li and Turner point out that it is also possible use the Stein gradient estimator with a model that admits a parametric density. This results in a smoothed version (loss uses RKHS norm rather than L_2 norm) of the original score matching estimator. Experimentally, in the HMC context, the methods seem to perform on-par in that case [75, Figure 1].

Bibliography

[1] M. Abadi et al. “Tensorflow: Large-scale machine learning on heterogeneous distributed systems”. In: *arXiv preprint arXiv:1603.04467* (2016).

[2] G. Alain and Y. Bengio. “What regularized auto-encoders learn from the data-generating distribution.” In: *JMLR* 15.1 (2014), pp. 3563–3593.

[3] N. Anderson, P. Hall, and D. Titterington. “Two-sample test statistics for measuring discrepancies between two multivariate probability density functions using kernel-based density estimates”. In: *Journal of Multivariate Analysis* 50 (1994), pp. 41–54.

[4] C. Andrieu and E. Moulines. “On the ergodicity properties of some adaptive MCMC algorithms”. In: *Ann. Appl. Probab.* 16.3 (Aug. 2006), pp. 1462–1505.

[5] C. Andrieu and G. Roberts. “The pseudo-marginal approach for efficient Monte Carlo computations”. In: *Annals of Statistics* 37.2 (2009), pp. 697–725.

[6] C. Andrieu and J. Thoms. “A tutorial on adaptive MCMC”. In: *Statistics and Computing* 18.4 (2008), pp. 343–373.

[7] J. Arvo. “Fast random rotation matrices”. In: *Graphics Gems III*. 1992.

[8] B. Schölkopf, A. J. Smola, and K.-R. Müller. “Nonlinear component analysis as a kernel Eigenvalue problem”. In: *Neural Comput.* 10 (5 1998), pp. 1299–1319.

[9] K. Bache and M. Lichman. *UCI Machine Learning Repository*. 2013. URL: `http://archive.ics.uci.edu/ml`.

[10] C. Baker. "Joint Measures and Cross-Covariance Operators". In: *Transactions of the American Mathematical Society* 186 (1973), pp. 273–289.

[11] G. Bakir, J. Weston, and B. Schölkopf. "Learning to find Pre-images". In: *Advances in Neural Information Processing Systems*. 2003.

[12] R. Bardenet, A. Doucet, and C. Holmes. "Towards scaling up Markov Chain Monte Carlo: an adaptive subsampling approach". In: *International Conference for Machine Learning*. 2014, pp. 405–413.

[13] L. Baringhaus and N. Henze. "A consistent test for multivariate normality based on the empirical characteristic function". In: *Metrika* 35 (1988), pp. 339–348.

[14] A. R. Barron. "Uniformly powerful goodness of fit tests." In: *The Annals of Statistics* 17 (1989), pp. 107–124.

[15] A. Barron and C.-H. Sheu. "Approximation of density functions by sequences of exponential families". In: *Annals of Statistics* 19.3 (1991), pp. 1347–1369.

[16] M. Beaumont. "Estimation of population growth or decline in genetically monitored populations". In: *Genetics* 164.3 (2003), pp. 1139–1160.

[17] J. Beirlant, L. Györfi, and G. Lugosi. "On the asymptotic normality of the L_1- and L_2-errors in histogram density estimation". In: *Canadian Journal of Statistics* 22 (1994), pp. 309–318.

[18] A. Ben-Israel and T. N. E. Greville. *Generalized inverses: theory and applications*. Second edition. Springer, 2003.

[19] A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer, 2004.

[20] M. Betancourt. "The Fundamental Incompatibility of Hamiltonian Monte Carlo and Data Subsampling". In: *arXiv preprint arXiv:1502.01510* (2015).

[21] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[22] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola. "Integrating structured biological data by Kernel Maximum Mean Discrepancy". In: *Bioinformatics (ISMB)* 22.14 (2006).

[23] S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford, UK: Oxford University Press, 2013.

[24] A. Bowman and P. Foster. "Adaptive smoothing and density based tests of multivariate normality". In: *J. Amer. Statist. Assoc* 88 (1993), pp. 529–537.

[25] L. D. Brown. *Fundamentals of Statistical Exponential Families with Applications in Statistical Decision Theory*. Hayward, CA: IMS, 1986.

[26] S. Canu and A. J. Smola. "Kernel methods and the exponential family". In: *Neurocomputing* 69.7 (2006), pp. 714–720.

[27] A. Caponnetto and E. De Vito. "Optimal rates for regularized least-squares algorithm". In: *Foundations of Computational Mathematics* 7.3 (2007), pp. 331–368.

[28] O. Cappé, a. Guillin, J. M. Marin, and C. P. Robert. "Population Monte Carlo". In: *Journal of Computational and Graphical Statistics* 13.4 (2004), pp. 907–929.

[29] O. Cappé, R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert. "Adaptive importance sampling in general mixture classes". In: *Statistics and Computing* 18.4 (2008), pp. 447–459.

[30] C. Carmeli, E. De Vito, A. Toigo, and V. Umanitá. "Vector valued reproducing kernel Hilbert spaces and universality". In: *Analysis and Applications* 8.01 (2010), pp. 19–61.

[31] T. Chen, E. Fox, and C. Guestrin. "Stochastic Gradient Hamiltonian Monte Carlo". In: *International Conference for Machine Learning*. 2014, pp. 1683–1691.

[32] N. Chopin. "A sequential particle filter method for static models". In: *Biometrika* 89.3 (2002), pp. 539–552.

[33] N. Chopin, P. E. Jacob, and O. Papaspiliopoulos. "SMC2: an efficient algorithm for sequential analysis of state space models". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 75.3 (2013), pp. 397–426.

[34] K. Chwialkowski, H. Strathmann, and A. Gretton. "A kernel test of goodness of fit". In: *International Conference for Machine Learning*. 2016.

[35] K. P. Chwialkowski, A. Ramdas, D. Sejdinovic, and A. Gretton. "Fast two-sample testing with analytic representations of probability measures". In: *Advances in Neural Information Processing Systems*. 2015, pp. 1981–1989.

[36] K. P. Chwialkowski, D. Sejdinovic, and A. Gretton. "A wild bootstrap for degenerate kernel tests". In: *Advances in Neural Information Processing Systems*. 2014, pp. 3608–3616.

[37] C. Cortes, M. Mohri, and A. Talwalkar. "On the impact of kernel approximation on learning accuracy". In: 2010.

[38] R. Douc and O. Cappé. "Comparison of resampling schemes for particle filtering". In: *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*. 2005.

[39] A. Doucet, N. D. Freitas, and N. Gordon. *An Introduction to Sequential Monte Carlo Methods*. Springer, 2001.

[40] A. Doucet and A. M. Johansen. "A tutorial on particle filtering and smoothing: Fifteen years later". In: *Handbook of nonlinear filtering* 12.656-704 (2009), p. 3.

[41] P. Drineas and M. W. Mahoney. "On the Nyström method for approximating a Gram matrix for improved kernel-based learning". In: *Journal of Machine Learning Research* 6 (2005), pp. 2153–2175.

[42] A. El Alaoui and M. W. Mahoney. "Fast Randomized Kernel Methods With Statistical Guarantees". In: *Advances in Neural Information Processing Systems*. 2015.

[43] P. Fearnhead and B. M. Taylor. "An Adaptive Sequential Monte Carlo Sampler". In: *Bayesian Analysis* 2 (2013), pp. 411–438.

[44] M. Filippone and M. Girolami. "Pseudo-marginal Bayesian inference for Gaussian Processes". In: *IEEE PAMI* (2014).

[45] M. Fromont, B. Laurent, M. Lerasle, and P. Reynaud-Bouret. "Kernels based tests with non-asymptotic bootstrap approaches for two-sample problems". In: 2012, pp. 23.1–23.22.

[46] K. Fukumizu, F. R. Bach, and M. I. Jordan. "Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces". In: *Journal of Machine Learning Research* 5 (2004), pp. 73–99.

[47] K. Fukumizu. "Exponential manifold by reproducing kernel Hilbert spaces". In: *Algebraic and Geometric Methods in Statistics*. Cambridge University Press, 2009, pp. 291–306.

[48] A. Gelman, G. O. Roberts, and W. R. Gilks. "Efficient Metropolis jumping rules". In: *Bayesian statistics*. 1996, pp. 599–607.

[49] A. Gelman and D. Rubin. "Inference from iterative simulation using multiple sequences". In: *Statistical science* (1992), pp. 457–472.

[50] M. Girolami and B. Calderhead. "Riemann manifold Langevin and Hamiltonian Monte Carlo methods". In: *Journal of the Royal Statistical Society: Series B* 73.2 (2011), pp. 123–214.

[51] G. H. Givens and A. E. Raftery. "Local adaptive importance sampling for multivariate densities with strong nonlinear relationships". In: *Journal of the American Statistical Association* 91.433 (1996), pp. 132–141.

[52] J. Gorham and L. Mackey. "Measuring Sample Quality with Stein's Method". In: *Advances in Neural Information Processing Systems*. 2015, pp. 226–234.

[53] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. "A Kernel Method for the Two-Sample Problem". In: *Advances in Neural Information Processing Systems*. 2007, pp. 513–520.

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

[55] A. Gretton, K. Fukumizu, C. Teo, L. Song, B. Schölkopf, and A. Smola. "A Kernel Statistical Test of Independence." In: *Advances in Neural Information Processing Systems*. Vol. 20. 2007, pp. 585–592.

[56] A. Gretton and L. Gyorfi. "Consistent Nonparametric Tests of Independence". In: *Journal of Machine Learning Research* 11 (2010), pp. 1391–1423.

[57] C. Gu and C. Qiu. "Smoothing spline density estimation: Theory". In: *Annals of Statistics* 21.1 (1993), pp. 217–234.

[58] L. Györfi and E. C. van der Meulen. "A consistent goodness of fit test based on the total variation distance". In: *Nonparametric Functional Estimation and Related Topics*. Ed. by G. Roussas. Kluwer, Dordrecht, 1990, pp. 631–645.

[59] L. Györfi and I. Vajda. "Asymptotic distributions for goodness of fit statistics in a sequence of multinomial models". In: *Statistics and Probability Letters* 56 (2002), pp. 57–67.

[60] H. Haario, E. Saksman, and J. Tamminen. "Adaptive Proposal Distribution for Random Walk Metropolis Algorithm". In: *Compututational Statistics* 14.3 (1999), pp. 375–395.

[61] H. Haario, E. Saksman, and J. Tamminen. "An adaptive Metropolis algorithm". In: *Bernoulli* 7.2 (2001), pp. 223–242.

[62] N. Hansen, S. D. Müller, and P. Koumoutsakos. "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)". In: *Evolutionary computation* 11.1 (2003), pp. 1–18.

[63] A. Hyvärinen. "Estimation of non-normalized statistical models by score matching". In: *Journal of Machine Learning Research* 6 (2005), pp. 695–709.

[64] A. Hyvärinen. "Some extensions of score matching". In: *Computational Statistics & Data Analysis* 51 (2007), pp. 2499–2512.

[65] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky. "Nonparametric belief propagation for self-localization of sensor networks". In: *IEEE Journal on selected Areas in Communications* 23.4 (2005), pp. 809–819.

[66] W. Jitkrittum, W. Xu, Z. Szabo, K. Fukumizu, and A. Gretton. "A Linear-Time Kernel Goodness-of-Fit Test". In: *arXiv preprint arXiv:1705.07673* (2017).

[67] A. Kolmogorov. "Sulla determinazione empirica di una legge di distribuzione". In: *G. Ist. Ital. Attuari* 4 (1933), pp. 83–91.

[68] A. Korattikara, Y. Chen, and M. Welling. "Austerity in MCMC Land: Cutting the Metropolis-Hastings Budget". In: *International Conference for Machine Learning*. 2014, pp. 181–189.

[69] A. Korattikara, Y. Chen, and M. Welling. "Austerity in MCMC land: Cutting the Metropolis-Hastings budget". In: *arXiv preprint arXiv:1304.5299* (2013).

[70] S. Lan, J. Streets, and B. Shahbaba. "Wormhole Hamiltonian Monte Carlo". In: *AAAI*. 2014.

[71] Q. Le, T. Sarlós, and A. Smola. "Fastfood–Approximating kernel expansions in loglinear time". In: *International Conference for Machine Learning*. 2013.

[72] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. J. Huang. "A Tutorial on Energy-Based Learning". In: *Predicting Structured Data*. MIT Press, 2006, pp. 191–246.

[73] A. Leucht. "Degenerate U- and V-statistics under weak dependence: Asymptotic theory and bootstrap consistency". In: *Bernoulli* 18.2 (2012), pp. 552–585.

[74] A. Leucht and M. Neumann. "Dependent wild bootstrap for degenerate U- and V-statistics". In: *Journal of Multivariate Analysis* 117 (2013), pp. 257–280.

[75] Y. Li and R. E. Turner. "Gradient Estimators for Implicit Models". In: *arXiv preprint arXiv:1705.07107* (2017).

[76] M. Lichman. *UCI Machine Learning Repository*. 2013. URL: `http://archive.ics.uci.edu/ml`.

[77] Q. Liu, J. Lee, and M. I. Jordan. "A Kernelized Stein Discrepancy for Goodness-of-fit Tests and Model Evaluation". In: *International Conference for Machine Learning*. 2016.

[78] Q. Liu and D. Wang. "Stein variational gradient descent: A general purpose bayesian inference algorithm". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2378–2386.

[79] J. R. Lloyd and Z. Ghahramani. "Statistical model criticism using kernel two sample tests". In: *Advances in Neural Information Processing Systems*. 2015, pp. 829–837.

[80] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré. "Markov chain Monte Carlo without likelihoods". In: *Proceedings of the National Academy of Sciences* 100.26 (2003), pp. 15324–15328.

[81] T. Marshall and G. Roberts. "An adaptive approach to Langevin MCMC". In: *Statistics and Computing* 22.5 (2012), pp. 1041–1057.

[82] E. Meeds, R. Leenders, and M. Welling. "Hamiltonian ABC". In: *Conference on Uncertainty in Artificial Intelligence*. 2015.

[83] K. Mengersen and R. Tweedie. "Rates of convergence of the Hastings and Metropolis algorithms". In: *The Annals of Statistics* 24.1 (1996), pp. 101–121.

[84] I. Murray and R. Adams. "Slice sampling covariance hyperparameters of latent Gaussian models". In: *Advances in Neural Information Processing Systems*. 2012.

[85] C. A. Naesseth and F. Lindsten. "Importance sampling with Hamiltonian dynamics". In: *Poster at NIPS Workshop 'Scalable Monte Carlo'*. 2015.

[86] R. Neal. "MCMC using Hamiltonian dynamics". In: *Handbook of Markov Chain Monte Carlo* 2.11 (2011).

[87] C. Nemeth, F. Lindsten, M. Filippone, and J. Hensman. "Pseudo-extended Markov chain Monte Carlo". In: *arXiv preprint arXiv:1708.05239* (2017).

[88] C. J. Oates, M. Girolami, and N. Chopin. "Control functionals for Monte Carlo integration". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79.3 (2017), pp. 695–718.

[89] A. Rahimi and B. Recht. "Random features for large-scale kernel machines". In: *Advances in Neural Information Processing Systems*. 2007, pp. 1177–1184.

[90] C. Rasmussen. "Gaussian Processes to Speed up Hybrid Monte Carlo for Expensive Bayesian Integrals". In: *Bayesian Statistics 7* (2003), pp. 651–659.

[91] M. L. Rizzo. "New Goodness-of-Fit Tests for Pareto Distributions". In: *ASTIN Bulletin: Journal of the International Association of Actuaries* 39.2 (2009), pp. 691–715.

[92] H. Robbins and S. Monro. "A stochastic approximation method". In: *The annals of mathematical statistics* (1951), pp. 400–407.

[93] C. P. Robert and C. George. *Monte carlo methods*. Springer.

[94] G. Roberts and J. Rosenthal. "Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms". In: *J. Appl. Probab.* 44.2 (Mar. 2007), pp. 458–475.

[95] G. Roberts and O. Stramer. "Langevin diffusions and Metropolis-Hastings algorithms". In: *Methodol. Comput. Appl. Probab.* 4 (2003), pp. 337–358.

[96] G. Roberts and R. Tweedie. "Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms". In: *Biometrika* 83.1 (1996), pp. 95–110.

[97] J. S. Rosenthal. "Optimal Proposal Distributions and Adaptive MCMC". In: *Handbook of Markov Chain Monte Carlo*. 2011. Chap. 4, pp. 93–112.

[98] A. Rudi, R. Camoriano, and L. Rosasco. *Generalization Properties of Learning with Random Features*. 2016.

[99] A. Rudi, R. Camoriano, and L. Rosasco. "Less is More: Nyström Computational Regularization". In: *Advances in Neural Information Processing Systems*. 2015.

[100] B. Schölkopf, R. Herbrich, and A. J. Smola. "A Generalized Representer Theorem". In: 2001.

[101] I. Schuster, H. Strathmann, B. Paige, and D. Sejdinovic. "Kernel Adaptive Sequential Monte Carlo". In: *European conference on machine learning & principles and practice of knowledge discovery in databases*. Joint first two authors. 2017.

[102] I. Schuster. "Gradient Importance Sampling". In: *arXiv preprint arXiv:1507.05781* (2015).

[103] D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu. "Equivalence of distance-based and RKHS-based statistics in hypothesis testing". In: 41.5 (2013), pp. 2263–2291.

[104] D. Sejdinovic, H. Strathmann, M. Lomeli, C. Andrieu, and A. Gretton. "Kernel Adaptive Metropolis-Hastings". In: *International Conference for Machine Learning*. 2012.

[105] "Sequential Monte Carlo samplers". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.3 (2006), pp. 411–436.

[106] R. Serfling. *Approximation Theorems of Mathematical Statistics*. New York: Wiley, 1980.

[107] X. Shao. "The Dependent Wild Bootstrap". In: 105.489 (2010), pp. 218–235.

[108] S. Sisson and Y. Fan. "Likelihood-free Markov chain Monte Carlo". In: *Handbook of Markov chain Monte Carlo* (2010).

[109] N. Smirnov. "Table for estimating the goodness of fit of empirical distributions". In: *Annals of Mathematical Statistics* 19 (1948), pp. 279–281.

[110] A. J. Smola, S. Mika, B. Schölkopf, and R. C. Williamson. "Regularized principal manifolds". In: *Journal of Machine Learning Research* 1 (2001), pp. 179–209.

[111] A. Smola, A. Gretton, L. Song, and B. Schölkopf. "A Hilbert Space Embedding for Distributions". In: *ALT*. 2007, pp. 13–31.

[112] A. J. Smola and B. Schölkopf. "Sparse Greedy Matrix Approximation for Machine Learning". In: *International Conference for Machine Learning*. 2000.

[113] J. Snoek, H. Larochelle, and R. P. Adams. "Practical bayesian optimization of machine learning algorithms". In: *Advances in Neural Information Processing Systems*. 2012.

[114] B. Sriperumbudur. "Mixture density estimation via Hilbert space embedding of measures". In: *Proceedings of the International Symposium on Information Theory*. 2011, pp. 1027–1030.

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

[116] B. K. Sriperumbudur and Z. Szábo. "Optimal rates for random Fourier features". In: *Advances in Neural Information Processing Systems*. 2015.

[117] B. Sriperumbudur, K. Fukumizu, A. Gretton, A. Hyvärinen, and R. Kumar. "Density estimation in infinite dimensional exponential families". In: *Journal of Machine Learning Research* 18.57 (2017), pp. 1–59.

[118] C. Stein. "A bound for the error in the normal approximation to the distribution of a sum of dependent random variables". In: *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and*

Probability, Volume 2: Probability Theory. Berkeley, Calif.: University of California Press, 1972, pp. 583–602.

[119] I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, 2008.

[120] H. Strathmann, D. Sejdinovic, S. Livingstone, Z. Szabo, and A. Gretton. "Gradient-free Hamiltonian Monte Carlo with Efficient Kernel Exponential Families". In: *Advances in Neural Information Processing Systems*. 2015.

[121] S. Sun, J. Xu, et al. "Learning structured densities via infinite dimensional exponential families". In: *Advances in Neural Information Processing Systems*. 2015, pp. 2287–2295.

[122] D. Sutherland, H. Strathmann, M. Arbel, and A. Gretton. "Efficient and principled score estimation". In: *arXiv preprint arXiv:1705.08360* (2017). Joint first two authors. Submitted.

[123] D. J. Sutherland and J. Schneider. "On the Error of Random Fourier Features". In: *Conference on Uncertainty in Artificial Intelligence*. 2015.

[124] G. J. Székely and M. L. Rizzo. "A new test for multivariate normality". In: *J. Multivariate Analysis* 93.1 (2005), pp. 58–80.

[125] M.-N. Tran, M. Scharth, M. K. Pitt, and R. Kohn. "Importance sampling squared for Bayesian inference in latent variable models". In: *arXiv preprint arXiv:1309.3339* (2013).

[126] M. J. Wainwright and M. I. Jordan. "Graphical models, exponential families, and variational inference". In: *Foundations and Trends® in Machine Learning* 1.1–2 (2008), pp. 1–305.

[127] L. Wasserman. *All of nonparametric statistics*. Springer, 2006.

[128] M. Welling and Y. Teh. "Bayesian Learning via Stochastic Gradient Langevin Dynamics". In: *International Conference for Machine Learning*. 2011, pp. 681–688.

[129] M. Welling and Y. Teh. "Bayesian learning via stochastic gradient Langevin dynamics". In: *International Conference for Machine Learning*. 2011, pp. 681–688.

[130] C. K. I. Williams and M. Seeger. "Using the Nyström method to speed up kernel machines". In: *Advances in Neural Information Processing Systems*. 2000.

[131] C. Williams and D. Barber. "Bayesian classification with Gaussian processes". In: *IEEE PAMI* 20.12 (1998), pp. 1342–1351.

[132] S. N. Wood. "Statistical inference for noisy nonlinear ecological dynamic systems". In: *Nature* 466.7310 (Aug. 2010), pp. 1102–1104.

[133] D. P. Woodruff. "Sketching as a Tool for Numerical Linear Algebra". In: *Foundations and Trends in Theoretical Computer Science* 10.1–2 (2014), pp. 1–157.

[134] Y. Yang, M. Pilanci, M. J. Wainwright, et al. "Randomized sketches for kernels: Fast and optimal nonparametric regression". In: *The Annals of Statistics* 45.3 (2017), pp. 991–1023.

[135] M. Yu, M. Kolar, and V. Gupta. "Statistical Inference for Pairwise Graphical Models Using Score Matching". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2829–2837.

[136] C. Zhang, B. Shahbaba, and H. Zhao. "Hamiltonian Monte Carlo acceleration using surrogate functions with random bases". In: *Statistics and Computing* (2016).

[137] C. Zhang, B. Shahbaba, and H. Zhao. "Variational Hamiltonian Monte Carlo via Score Matching". In: *Bayesian Analysis* (2017).