Performance analysis of greedy algorithms for minimising a Maximum Mean Discrepancy

Luc Pronzato

Performance analysis of greedy algorithms for minimising a Maximum Mean Discrepancy

Luc Pronzato

Université Côte d'Azur, CNRS, Laboratoire I3S
Bât. Euclide, Les Algorithmes, 2000 route des lucioles,
06900 Sophia Antipolis, France
Luc.Pronzato@cnrs.fr

April 28, 2022

Abstract

We analyse the performance of several iterative algorithms for the quantisation of a probability measure μ, based on the minimisation of a Maximum Mean Discrepancy (MMD). Our analysis includes kernel herding, greedy MMD minimisation and Sequential Bayesian Quadrature (SBQ). We show that the finite-sample-size approximation error, measured by the MMD, decreases as $1/n$ for SBQ and also for kernel herding and greedy MMD minimisation when using a suitable step-size sequence. The upper bound on the approximation error is slightly better for SBQ, but the other methods are significantly faster, with a computational cost that increases only linearly with the number of points selected. This is illustrated by two numerical examples, with the target measure μ being uniform (a space-filling design application) and with μ a Gaussian mixture. They suggest that the bounds derived in the paper are overly pessimistic, in particular for SBQ. The sources of this pessimism are identified but seem difficult to counter.

keywords Maximum Mean Discrepancy; quantisation; greedy algorithm; sequential Bayesian quadrature; kernel herding; space-filling design; computer experiments

1 Introduction and motivation

Background. Quantisation of a probability measure μ is a basic task in many fields, such as probabilistic integration (Briol et al., 2019), MCMC computation (Joseph et al., 2015a, 2019) or space-filling design in computer experiments (Joseph et al., 2015b; Mak and Joseph, 2017, 2018; Pronzato and Zhigljavsky, 2020), and minimisation of the Maximum Mean Discrepancy (MMD) defined by a kernel K is a powerful tool for this task[1]. In particular, it easily allows iterative constructions that can be stopped when the discrete approximation obtained is deemed sufficient, a situation where the number of support points is not fixed in advance.

[1]We do not consider quantisation methods based on Voronoi partitions, for which one can refer in particular to Graf and Luschgy (2000).

Claims and hint of the contents. We derive finite-sample-size errors bounds for iterative methods to quantise a probability measure by minimising the MMD for a given kernel. The methods considered include gradient-type algorithms (kernel herding), greedy one-step-ahead minimisation, and Sequential Bayesian Quadrature (SBQ) that sets optimal weights on the current support at each iteration. Two variants of SBQ are considered, with and without the constraint that the weights sum to one (the bound for the unconstrained version is markedly pessimistic but our analysis reveals a connection with kernel herding and gives some insight for the reason of this pessimism). We consider the practical situation where the candidate set is finite; it may correspond in particular to points independently sampled with μ, with the possibility to use a different set at every iteration (see Section 6 and Appendix C). The context of a finite candidate set is the most widely used in practical situations. It allows us to derive simple proofs that only use (finite-dimensional) linear algebra, but our results can be extended to the infinite-dimensional (Hilbert space) situation, where the new support point selected at each iteration is searched within a continuous set; see, e.g., Chen et al. (2018); Teymur et al. (2021). We show that the error is $\mathcal{O}(n^{-1})$ for SBQ and for algorithms that use a suitable step-size sequence and construct nonuniform discrete measures (with a slightly better constant for SBQ). We show that it is also $\mathcal{O}(n^{-1})$ for the construction of uniform (empirical) measures provided that the measure with total mass one minimising the MMD over the candidate set is a probability measure. We show that the complexity of gradient and greedy one-step-ahead methods grows linearly with n, whereas it grows quadratically for SBQ. Two variants of kernel herding are considered, with similar performance to SBQ but slightly lighter calculations.

Paper organisation. Section 2 recalls the background on MMD and Bayesian quadrature. It defines the notation and introduces the methods that are considered in the rest of the paper. The performance of kernel herding is analysed in Section 3. The results presented in Sections 3.1 and 3.2 are not new, but the analysis of this basic gradient-type algorithm is central to the investigation of the convergence rate for the other methods, more sophisticated, that we consider in Sections 3.3 (variants of kernel herding), 4 (greedy MMD minimisation) and 5 (SBQ). Section 6 extends the results of previous sections to the case where the candidate set corresponds to points independently sampled with μ. Two illustrative examples are presented in Section 7, one with μ uniform (space-filling design), the other with μ a Gaussian mixture. Section 8 concludes briefly.

2 Maximum Mean Discrepancy and Bayesian quadrature

2.1 Maximum Mean Discrepancy (MMD)

Let $\mathscr{X}$ be a measurable set, equipped with a probability measure μ. For instance, for application to space-filling design for computer experiments, $\mathscr{X}$ is typically a compact subset of $\mathbb{R}^d$ and μ is proportional to the Lebesgue measure on $\mathscr{X}$. Let K be a symmetric strictly positive definite (s.p.d.) kernel defined on $\mathscr{X} \times \mathscr{X}$, uniformly bounded on $\mathscr{X}$; that is,

$$K(\mathbf{x}, \mathbf{x}) \leq \overline{K} < +\infty\,, \quad \text{for all } \mathbf{x} \in \mathscr{X}\,, \tag{1}$$

and for any $n \in \mathbb{N}$ and any $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in $\mathscr{X}$, the $n \times n$ matrix $\mathbf{K}_n$ with element i, j equal to $K(\mathbf{x}_i, \mathbf{x}_j)$ is p.d. and s.p.d. when the $\mathbf{x}_i$ are all pairwise different. Note that K being s.p.d. implies that $K^2(\mathbf{x}, \mathbf{x}') \leq K(\mathbf{x}, \mathbf{x})K(\mathbf{x}', \mathbf{x}')$ for all $\mathbf{x}$ and $\mathbf{x}'$ in $\mathscr{X}$ with the inequality being strict

when $\mathbf{x} \neq \mathbf{x}'$. Moreover, (1) implies $\tau_\gamma(\mu) = \int_{\mathscr{X}} K^\gamma(\mathbf{x},\mathbf{x})\,\mathrm{d}\mu(\mathbf{x}) \leq \overline{K}^\gamma < +\infty$ for any $\gamma \geq 0$. K defines a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_K$, and we respectively denote by $\langle\cdot,\cdot\rangle_K$ and $\|\cdot\|_{\mathcal{H}_K}$ the scalar product and norm in $\mathcal{H}_K$. We do not assume that $\mathcal{H}_K$ is finite-dimensional. We say that K is positive ($K \geq 0$) when $K(\mathbf{x},\mathbf{x}') \geq 0$ for all $\mathbf{x}$ and $\mathbf{x}'$ in $\mathscr{X}$.

We denote by $\mathscr{M}(\mathscr{X})$ the set of finite signed measures on $\mathscr{X}$, by $\mathscr{M}_{[1]}(\mathscr{X})$ the set of signed measures with total mass 1, and by $\mathscr{M}^+_{[1]}(\mathscr{X})$ the set of probability measures on $\mathscr{X}$ (with thus $\mu \in \mathscr{M}^+_{[1]}(\mathscr{X})$). The reproducing property implies that, for any $\nu \in \mathscr{M}(\mathscr{X})$, the *energy* of ν, defined by $\mathscr{E}_K(\nu) = \int_{\mathscr{X}^2} K(\mathbf{x},\mathbf{x}')\,\mathrm{d}\nu(\mathbf{x})\,\mathrm{d}\nu(\mathbf{x}')$, satisfies

$$\begin{aligned}\mathscr{E}_K(\nu) = \int_{\mathscr{X}^2} \langle K(\mathbf{x},\cdot), K(\mathbf{x}',\cdot)\rangle_K\,\mathrm{d}\nu(\mathbf{x})\,\mathrm{d}\nu(\mathbf{x}') &\leq \int_{\mathscr{X}^2} \|K(\mathbf{x},\cdot)\|_{\mathcal{H}_K}\,\|K(\mathbf{x}',\cdot)\|_{\mathcal{H}_K}\,\mathrm{d}\nu(\mathbf{x})\,\mathrm{d}\nu(\mathbf{x}') \\ &= \left[\int_{\mathscr{X}} K^{1/2}(\mathbf{x},\mathbf{x})\,\mathrm{d}\nu(\mathbf{x})\right]^2 = \tau^2_{1/2}(\nu) < +\infty\,.\end{aligned}$$

For any $\nu \in \mathscr{M}(\mathscr{X})$ and any $\mathbf{x} \in \mathscr{X}$, we denote by

$$P_{K,\nu}(\mathbf{x}) = \int_{\mathscr{X}} K(\mathbf{x},\mathbf{x}')\,\mathrm{d}\nu(\mathbf{x}')$$

the *potential* of ν at $\mathbf{x}$; $P_{K,\nu}(\cdot)$ is also called the *kernel imbedding* of ν into $\mathcal{H}_K$.

For μ and ν in $\mathscr{M}^+_{[1]}(\mathscr{X})$, any $f \in \mathcal{H}_K$ satisfies the following (Koksma-Hlawka type) inequality: $\left|\int_{\mathscr{X}} f(\mathbf{x})\,\mathrm{d}\nu(\mathbf{x}) - \int_{\mathscr{X}} f(\mathbf{x})\,\mathrm{d}\mu(\mathbf{x})\right| = |\langle f, P_{K,\mu} - P_{K,\nu}\rangle_K| \leq \|f\|_{\mathcal{H}_K}\,\mathsf{MMD}_K(\mu,\nu)$, where

$$\begin{aligned}\mathsf{MMD}_K(\mu,\nu) &= \sup_{\|f\|_{\mathcal{H}_K}=1} \left|\int_{\mathscr{X}} f(\mathbf{x})\,\mathrm{d}\nu(\mathbf{x}) - \int_{\mathscr{X}} f(\mathbf{x})\,\mathrm{d}\mu(\mathbf{x})\right| = \|P_{K,\nu} - P_{K,\mu}\|_{\mathcal{H}_K} \\ &= \mathscr{E}_K^{1/2}(\nu-\mu) = \left[\int_{\mathscr{X}^2} K(\mathbf{x},\mathbf{x}')\,\mathrm{d}(\nu-\mu)(\mathbf{x})\,\mathrm{d}(\nu-\mu)(\mathbf{x}')\right]^{1/2} \\ &= \left[\mathscr{E}_K(\mu) + \mathscr{E}_K(\nu) - 2\int_{\mathscr{X}} P_{K,\mu}(\mathbf{x})\,\mathrm{d}\nu(\mathbf{x})\right]^{1/2} \qquad (2)\end{aligned}$$

is called the *Maximum-Mean-Discrepancy* (MMD) between ν and μ; see Sejdinovic et al. (2013, Def. 10). $\mathsf{MMD}_K(\mu,\nu)$ defines an integral pseudometric between probability distributions and a pseudometric between kernel imbeddings. It defines a metric on $\mathscr{M}^+_{[1]}(\mathscr{X})$ when K is characteristic[2], which we assume in the following. This implies in particular that $\mathsf{MMD}_K(\mu,\nu) > 0$ for any $\nu \in \mathscr{M}_{[1]}(\mathscr{X})$, $\nu \neq \mu$.

For a collection $\mathbf{X}_n = \{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ of n points in $\mathscr{X}$, called n-point design, we denote by $\xi_{n,e} = (1/n)\sum_{i=1}^n \delta_{\mathbf{x}_i}$ the associated empirical measure, with $\delta_{\mathbf{x}}$ the Dirac delta measure at $\mathbf{x}$. For $\mathbf{w}_n = (\{\mathbf{w}_n\}_1,\ldots,\{\mathbf{w}_n\}_n)^\top \in \mathbb{R}^n$ a vector of n weights, we denote by $\xi_n = \xi(\mathbf{w}_n)$ the signed measure

$$\xi(\mathbf{w}_n) = \sum_{i=1}^n \{\mathbf{w}_n\}_i\,\delta_{\mathbf{x}_i} \qquad (3)$$

[2]Since K is uniformly bounded, this is equivalent to the condition that K be Conditionally Integrally Strictly Positive Definite (CISPD), that is, $\mathscr{E}_K(\nu) > 0$ for all nonzero signed measure $\nu \in \mathscr{M}_{[0]}(\mathscr{X})$; see Sriperumbudur et al. (2010, Def. 6 and Lemma 8); see also Pronzato and Zhigljavsky (2020) for a comprehensive survey including the case of singular kernels.

(so that $\xi_{n,e} = \xi(\mathbf{1}_n/n)$, with $\mathbf{1}_n$ the n-dimensional vector with all components equal to 1). An important area of application for MMD minimisation is space-filling design, where the objective is to build evenly distributed designs on a compact $\mathscr{X}$; see, for example, Pronzato and Müller (2012); Pronzato (2017). Minimising $\mathsf{MMD}_K(\mu, \xi_{n,e})$ with μ uniform over $\mathscr{X}$ is then an effective approach to achieve this goal. One may also minimise $\mathsf{MMD}_K(\mu, \xi(\mathbf{w}_n))$ with respect to $\mathbf{X}_n$ and $\mathbf{w}_n$, and the designs obtained differ depending on the chosen kernel K, the constraints set on $\mathbf{w}_n$ and on the optimisation method that is used. In this paper, we focuss our attention on the construction of extensive point sequences $\mathbf{X}_n = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, such that $\mathbf{X}_{k+1} = [\mathbf{X}_k, \mathbf{x}_{k+1}]$ for all k, with the property that $\mathbf{X}_n$ is the support of a measure ξ_n which approximates μ well in the sense of $\mathsf{MMD}_K(\mu, \xi_n)$.

Analytic expressions for the quantities $\mathscr{E}_K(\mu)$ and $P_{K,\mu}(\cdot)$ that appear in (2) are available for particular measures and particular kernels, see Table 1 of Briol et al. (2019). This includes the case when μ is uniform on $\mathscr{X} = [0,1]^d$ and K is separable, see for example Table 3.1 of (Pronzato and Zhigljavsky, 2020), and separable kernels K based on variants of Brownian motion covariance yield L_2 discrepancies (symmetric, centred, wrap-around and so on); see Hickernell (1998), Fang et al. (2006, Chap. 3). $\mathscr{E}_K(\mu)$ and $P_{K,\mu}(\cdot)$ are not available when μ is a posterior distribution with unknown normalising constant; in that case, Joseph et al. (2015a, 2019) suggest to construct minimum-energy designs that minimise $\mathscr{E}_K(\xi_{n,e})$ for a particular kernel K. Another way is to minimise a kernel Stein discrepancy, that is, to minimise MMD for the image K' of a kernel K under a Stein operator, so that $\mathscr{E}_{K'}(\mu) = 0$ and $P_{K',\mu}(\mathbf{x}) = 0$ for any $\mathbf{x}$; see Chen et al. (2018); Detommaso et al. (2018); Gorham and MacKey (2017); Liu and Wang (2016); Oates et al. (2017). Throughout the paper we consider the general framework where $\mathcal{H}_K$ is an infinite-dimensional RKHS and assume that $\mathscr{E}_K(\mu)$ and $P_{K,\mu}(\mathbf{x})$ can be easily computed for any $\mathbf{x} \in \mathscr{X}$ (Monte-Carlo methods can always be used as a last resort).

2.2 MMD and optimal weights for discrete measures

For a given design $\mathbf{X}_n$, $\mathsf{MMD}_K(\mu, \xi_n)$ is quadratic in $\mathbf{w}_n$, and the optimal weights are easily obtained. Indeed, (2) gives

$$\begin{aligned} \mathsf{MMD}_K^2(\mu, \xi_n) = \mathscr{E}_K(\xi_n - \mu) &= \sum_{i,j=1}^{n} \{\mathbf{w}_n\}_i \{\mathbf{w}_n\}_j K(\mathbf{x}_i, \mathbf{x}_j) - 2 \sum_{i=1}^{n} \{\mathbf{w}_n\}_i P_{K,\mu}(\mathbf{x}_i) + \mathscr{E}_K(\mu)\,, \\ &= \mathbf{w}_n^\top \mathbf{K}_n \mathbf{w}_n - 2\, \mathbf{w}_n^\top \mathbf{p}_n(\mu) + \mathscr{E}_K(\mu)\,, \end{aligned} \tag{4}$$

where $\mathbf{p}_n(\mu) = [P_{K,\mu}(\mathbf{x}_1), \ldots, P_{K,\mu}(\mathbf{x}_n)]^\top$ (alternative expressions for $\mathsf{MMD}_K^2(\mu, \xi_n)$ are given in Appendix A). Therefore, $\mathbf{w}_n^*$ that minimises $\mathsf{MMD}_K^2(\mu, \xi_n)$ under the constraints $\{\mathbf{w}_n\}_i \geq 0$ and $\mathbf{1}_n^\top \mathbf{w}_n = 1$ is solution of a Quadratic Programming (QP) problem. We assume that the $\mathbf{x}_i$ in $\mathbf{X}_n$ are pairwise different, so that $\mathbf{K}_n$ has full rank. Releasing the positivity constraints, $\widehat{\mathbf{w}}_n$ that minimises $\mathsf{MMD}_K^2(\mu, \xi_n)$ with $\mathbf{1}_n^\top \mathbf{w}_n = 1$ is obtained explicitly as

$$\widehat{\mathbf{w}}_n = \left(\mathbf{K}_n^{-1} - \frac{\mathbf{K}_n^{-1} \mathbf{1}_n \mathbf{1}_n^\top \mathbf{K}_n^{-1}}{\mathbf{1}_n^\top \mathbf{K}_n^{-1} \mathbf{1}_n} \right) \mathbf{p}_n(\mu) + \frac{\mathbf{K}_n^{-1} \mathbf{1}_n}{\mathbf{1}_n^\top \mathbf{K}_n^{-1} \mathbf{1}_n} \tag{5}$$

(with $\mathbf{w}_n^* = \widehat{\mathbf{w}}_n$ when all components of $\widehat{\mathbf{w}}_n$ are nonnegative). Also, the unconstrained weights that minimise $\mathsf{MMD}_K^2(\mu, \xi_n)$ are given by

$$\widetilde{\mathbf{w}}_n = \mathbf{K}_n^{-1} \mathbf{p}_n(\mu)\,. \tag{6}$$

Throughout the paper, for any measure ξ_n supported on $\mathbf{X}_n$, we denote by ξ_n^*, $\widehat{\xi}_n$ and $\widetilde{\xi}_n$ the measures with the same support and respective weights $\mathbf{w}_n^*$, $\widehat{\mathbf{w}}_n$ and $\widetilde{\mathbf{w}}_n$, so that $\mathsf{MMD}_K^2(\mu, \widetilde{\xi}_n) \leq \mathsf{MMD}_K^2(\mu, \widehat{\xi}_n) \leq \mathsf{MMD}_K^2(\mu, \xi_n^*)$ (and $\mathsf{MMD}_K^2(\mu, \xi_n^*) \leq \mathsf{MMD}_K^2(\mu, \xi_n)$ if $\xi_n \in \mathscr{M}_{[1]}^+(\mathbf{X}_n)$).

2.3 Incremental MMD minimisation

We consider three families of incremental constructions.

2.3.1 Sequential Bayesian Quadrature (SBQ)

The construction of a design $\mathbf{X}_n$ that minimises $\mathsf{MMD}_K^2(\mu, \widetilde{\xi}_n)$ or $\mathsf{MMD}_K^2(\mu, \widehat{\xi}_n)$ is called Bayesian quadrature (BQ); it can be Sequential (SBQ), see Briol et al. (2015), and we consider two versions of SBQ. Bounds on their finite-sample-size error are given in Section 5. Note that $\mathsf{MMD}_K^2(\mu, \widetilde{\xi}_k) = \mathsf{MMD}_K^2(\mu, \widetilde{\xi}_{k+1})$ (respectively, $\mathsf{MMD}_K^2(\mu, \widehat{\xi}_k) = \mathsf{MMD}_K^2(\mu, \widehat{\xi}_{k+1})$) when $\widetilde{\xi}_k$ and $\widetilde{\xi}_{k+1}$ (respectively, $\widehat{\xi}_k$ and $\widehat{\xi}_{k+1}$) have the same support, so that SBQ always selects new points whenever possible (i.e., until all eligible points are exhausted). We may thus assume that the $\mathbf{x}_i$ are all pairwise different, $i = 1, \dots, k$, and that $\mathbf{K}_k$ has full rank for all k.

***(i)* Greedy minimisation of $\mathsf{MMD}_K^2(\mu, \widetilde{\xi}_k)$.** The equations (4) and (6) give $\mathsf{MMD}_K^2(\mu, \widetilde{\xi}_k) = \mathscr{E}_K(\mu) - \mathbf{p}_k^\top(\mu)\mathbf{K}_k^{-1}\mathbf{p}_k(\mu)$. We have

$$\mathbf{K}_{n+1} = \begin{bmatrix} \mathbf{K}_n & \mathbf{k}_n(\mathbf{x}_{n+1}) \\ \mathbf{k}_n^\top(\mathbf{x}_{n+1}) & K(\mathbf{x}_{n+1}, \mathbf{x}_{n+1}) \end{bmatrix},$$

where $\mathbf{k}_n(\mathbf{x}) = [K(\mathbf{x}_1, \mathbf{x}), \dots, K(\mathbf{x}_n, \mathbf{x})]^\top$, $\mathbf{x} \in \mathscr{X}$. The calculation of its inverse by

$$\mathbf{K}_{n+1}^{-1} = \begin{pmatrix} \mathbf{K}_n^{-1} + \beta_{n+1}\, \mathbf{u}_{n+1}\mathbf{u}_{n+1}^\top & -\beta_{n+1}\, \mathbf{u}_{n+1} \\ -\beta_{n+1}\, \mathbf{u}_{n+1}^\top & \beta_{n+1} \end{pmatrix}, \tag{7}$$

with $\mathbf{u}_{n+1} = \mathbf{K}_n^{-1}\mathbf{k}_n(\mathbf{x}_{n+1})$ and $\beta_{n+1} = [K(\mathbf{x}_{n+1}, \mathbf{x}_{n+1}) - \mathbf{k}_n^\top(\mathbf{x}_{n+1})\mathbf{u}_{n+1}]^{-1}$, will be used several times for incremental constructions and gives here

$$\mathsf{MMD}_K^2(\mu, \widetilde{\xi}_{k+1}) = \mathsf{MMD}_K^2(\mu, \widetilde{\xi}_k) - \frac{\left[\mathbf{p}_k^\top(\mu)\mathbf{K}_k^{-1}\mathbf{k}_k(\mathbf{x}_{k+1}) - P_{K,\mu}(\mathbf{x}_{k+1})\right]^2}{K(\mathbf{x}_{k+1}, \mathbf{x}_{k+1}) - \mathbf{k}_k^\top(\mathbf{x}_{k+1})\mathbf{K}_k^{-1}\mathbf{k}_k(\mathbf{x}_{k+1})}.$$

Since $\widetilde{\mathbf{w}}_k$ satisfies (6), we get

$$\mathsf{MMD}_K^2(\mu, \widetilde{\xi}_{k+1}) = \mathsf{MMD}_K^2(\mu, \widetilde{\xi}_k) - \frac{\left[P_{K,\widetilde{\xi}_k}(\mathbf{x}_{k+1}) - P_{K,\mu}(\mathbf{x}_{k+1})\right]^2}{\min_{\mathbf{w}\in\mathbb{R}^k} \|K(\mathbf{x}_{k+1}, \cdot) - \mathbf{w}^\top\mathbf{k}_k(\cdot)\|_{\mathcal{H}_K}^2}. \tag{8}$$

This corresponds to the "standard" version of SBQ, which uses general signed measures $\widetilde{\xi}_k$ in $\mathscr{M}(\mathscr{X})$: it selects $\mathbf{x}_1 \in \operatorname{Arg\,max}_{\mathbf{x}\in\mathscr{X}} P_{K,\mu}^2(\mathbf{x})/K(\mathbf{x}, \mathbf{x})$ and then

$$\mathbf{x}_{k+1} \in \operatorname*{Arg\,max}_{\mathbf{x}\in\mathscr{X}} \frac{\left[P_{K,\widetilde{\xi}_k}(\mathbf{x}) - P_{K,\mu}(\mathbf{x})\right]^2}{K(\mathbf{x}, \mathbf{x}) - \mathbf{k}_k^\top(\mathbf{x})\mathbf{K}_k^{-1}\mathbf{k}_k(\mathbf{x})}, \ k \geq 1. \tag{9}$$

***(ii)* Greedy minimisation of $\mathsf{MMD}_K^2(\mu, \widehat{\xi}_k)$.** The equations (4) and (5) give

$$\mathsf{MMD}_K^2(\mu, \widehat{\xi}_k) \;=\; \mathscr{E}_K(\mu) - \mathbf{p}_k^\top(\mu)\mathbf{K}_k^{-1}\mathbf{p}_k(\mu) + \frac{(1 - \mathbf{p}_k^\top(\mu)\mathbf{K}_k^{-1}\mathbf{1}_k)^2}{\mathbf{1}_k^\top\mathbf{K}_k^{-1}\mathbf{1}_k}\,,$$

but a simpler expression can be obtained through the introduction of the reduced kernel K_μ defined by

$$K_\mu(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}, \mathbf{x}') - P_{K,\mu}(\mathbf{x}) - P_{K,\mu}(\mathbf{x}') + \mathscr{E}_K(\mu)\,,\ \mathbf{x}, \mathbf{x}' \in \mathscr{X}\,. \tag{10}$$

Let $\{\mathbf{K}_{\mu_k}\}_{i,j} = K_\mu(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, \ldots, k$, so that $\mathbf{K}_{\mu_k} = \mathbf{K}_k - \mathbf{p}_k(\mu)\mathbf{1}_k^\top - \mathbf{1}_k\mathbf{p}_k^\top(\mu) + \mathscr{E}_K(\mu)\,\mathbf{1}_k\mathbf{1}_k^\top$ and, for any measure ξ_k with total mass one,

$$\mathsf{MMD}_K^2(\mu, \xi_k) = \mathbf{w}_k^\top\mathbf{K}_{\mu_k}\mathbf{w}_k\,. \tag{11}$$

As K is characteristic and $\mathbf{K}_k$ has full rank, $\mathbf{K}_{\mu_k}$ is invertible when μ is not fully supported on $\mathbf{X}_k$. Indeed, let $\mathbf{u}_k$ be an eigenvector of $\mathbf{K}_{\mu_k}$. If $a = \mathbf{u}_k^\top\mathbf{1}_k \neq 0$, then the measure ξ_k with weights $\mathbf{w}_k = \mathbf{u}_k/a$ has total mass one and satisfies $\mathsf{MMD}_K^2(\mu, \xi_k) = \gamma_k > 0$ since $\xi_k \neq \mu$, so that $\mathbf{u}_k^\top\mathbf{K}_{\mu_k}\mathbf{u}_k = a^2\gamma_k > 0$. If $a = \mathbf{u}_k^\top\mathbf{1}_k = 0$, then $\mathbf{u}_k^\top\mathbf{K}_{\mu_k}\mathbf{u}_k = \mathbf{u}_k^\top\mathbf{K}_k\mathbf{u}_k$, which is strictly positive since $\mathbf{K}_k$ has full rank. Direct calculation then gives

$$\widehat{\mathbf{w}}_k = \mathbf{K}_{\mu_k}^{-1}\mathbf{1}_k/(\mathbf{1}_k^\top\mathbf{K}_{\mu_k}^{-1}\mathbf{1}_k) \;\text{ and }\; \mathsf{MMD}_K^2(\mu, \widehat{\xi}_k) = 1/(\mathbf{1}_k^\top\mathbf{K}_{\mu_k}^{-1}\mathbf{1}_k)\,,$$

and, using block matrix inversion for $\mathbf{K}_{\mu_{k+1}}$,

$$\mathsf{MMD}_K^2(\mu, \widehat{\xi}_{k+1}) = \left\{\mathbf{1}_k^\top\mathbf{K}_{\mu_k}^{-1}\mathbf{1}_k + \frac{\left[\mathbf{1}_k^\top\mathbf{K}_{\mu_k}^{-1}\mathbf{k}_{\mu_k}(\mathbf{x}_{k+1}) - 1\right]^2}{K_\mu(\mathbf{x}_{k+1}, \mathbf{x}_{k+1}) - \mathbf{k}_{\mu_k}^\top(\mathbf{x}_{k+1})\mathbf{K}_{\mu_k}^{-1}\mathbf{k}_{\mu_k}(\mathbf{x}_{k+1})}\right\}^{-1}\,,$$

where $\mathbf{k}_{\mu_k}(\mathbf{x}) = [K_\mu(\mathbf{x}_1, \mathbf{x}), \ldots, K_\mu(\mathbf{x}_k, \mathbf{x})]^\top$, $\mathbf{x} \in \mathscr{X}$. Straightforward manipulations using (5) give

$$\mathsf{MMD}_K^2(\mu, \widehat{\xi}_{k+1}) = \mathsf{MMD}_K^2(\mu, \widehat{\xi}_k) - \frac{\left[P_{K,\widehat{\xi}_k}(\mathbf{x}_{k+1}) - P_{K,\mu}(\mathbf{x}_{k+1}) + \widehat{\mathbf{w}}_k^\top\mathbf{p}_k(\mu) - \mathscr{E}_K(\widehat{\xi}_k)\right]^2}{\min\limits_{\substack{\mathbf{w}\,\in\,\mathbb{R}^k \\ \mathbf{1}_k^\top\mathbf{w}\,=\,1}} \|K(\mathbf{x}_{k+1}, \cdot) - \mathbf{w}^\top\mathbf{k}_k(\cdot)\|_{\mathcal{H}_K}^2}\,. \tag{12}$$

This version of SBQ selects $\mathbf{x}_1 \in \operatorname{Arg\,min}_{\mathbf{x}\in\mathscr{X}} K_\mu(\mathbf{x}, \mathbf{x})$ and then

$$\mathbf{x}_{k+1} \in \operatorname*{Arg\,max}_{\mathbf{x}\in\mathscr{X}} \frac{\left[P_{K,\widehat{\xi}_k}(\mathbf{x}_) - P_{K,\mu}(\mathbf{x}) + \widehat{\mathbf{w}}_k^\top\mathbf{p}_k(\mu) - \widehat{\mathbf{w}}_k^\top\mathbf{K}_k\widehat{\mathbf{w}}_k)\right]^2}{K(\mathbf{x}, \mathbf{x}) - \mathbf{k}_k^\top(\mathbf{x})\mathbf{K}_k^{-1}\mathbf{k}_k(\mathbf{x}) + \frac{[1 - \mathbf{1}_k^\top\mathbf{K}_k^{-1}\mathbf{k}_k(\mathbf{x})]^2}{\mathbf{1}_k^\top\mathbf{K}_k^{-1}\mathbf{1}_k}}\,,\ k \geq 1\,. \tag{13}$$

The expressions (8) and (12) are pivotal to the derivation of finite-sample-size error bounds for SBQ through the consideration of simplified versions where $\mathbf{x}_{k+1}$ is chosen by kernel herding, see Section 3.3.1. When $\mathbf{x}$ is selected at each iteration within a candidate set of size C, see Section 2.4, the complexity of SBQ grows like $\mathcal{O}(n^2\,C)$ for n iterations (the main contribution comes from the denominators in (9) and (13) which must be calculated for the C candidates, but matrix-vector multiplications $\mathbf{K}_k^{-1}\mathbf{k}_k(\mathbf{x})$ can be avoided by using recursive calculation; see Remark 4).

2.3.2 Greedy MMD Minimisation (GM)

To lighten the computations required by SBQ, we can consider the optimal choice of successive $\mathbf{x}_k$ for a predefined sequence of weights $\mathbf{w}_k$. The standard version of Greedy MMD Minimisation (GM) uses $\mathbf{w}_k = \mathbf{1}_k/k$ for all k, so that ξ_k is the empirical measure $\xi_{k,e}$ supported on $\mathbf{X}_k$. It selects $\mathbf{x}_1 \in \operatorname{Arg\,min}_{\mathbf{x}\in\mathscr{X}} K(\mathbf{x},\mathbf{x}) - 2\,P_{K,\mu}(\mathbf{x}) = \operatorname{Arg\,min}_{\mathbf{x}\in\mathscr{X}} K_\mu(\mathbf{x},\mathbf{x})$ and then minimises $\mathsf{MMD}_K(\mu,\xi_{k+1,e})$ incrementally: (4) gives

$$\mathbf{x}_{k+1} \in \operatorname{Arg\,min}_{\mathbf{x}\in\mathscr{X}} \sum_{i=1}^{k} K(\mathbf{x}_i,\mathbf{x}) + \frac{1}{2}\,K(\mathbf{x},\mathbf{x}) - (k+1)\,P_{K,\mu}(\mathbf{x})\,,\ k \geq 1\,. \tag{14}$$

GM will be considered in Section 4. The complexity of MMD grows linearly and is $\mathcal{O}(n\,C)$ for n iterations when the selection is among C possible candidates. In Section 4.2 we also consider versions with nonuniform weights: one must then define the weight w_{k+1} to be allocated to the next point $\mathbf{x}_{k+1}$, not selected yet. A convenient way to proceed, usual in the area of optimal design of experiments, is to take $\xi_{k+1} = (1-\alpha_{k+1})\,\xi_k + \alpha_{k+1}\delta_{\mathbf{x}_{k+1}}$, for some step size $\alpha_{k+1} \in [0,1]$. This construction guarantees that $\xi_{k+1} \in \mathscr{M}^+_{[1]}(\mathscr{X})$ when $\xi_k \in \mathscr{M}^+_{[1]}(\mathscr{X})$; the choice of α_{k+1} defines the sequence of weights $\mathbf{w}_k$ and the point $\mathbf{x}_{k+1}$ is chosen to minimise $\mathsf{MMD}_K(\mu,\xi_{k+1})$.

2.3.3 The Frank-Wolfe algorithm and Kernel Herding (KH)

Another way of proceeding consists in exploiting the convexity of the functional $\phi_{K,\mu}(\cdot) : \xi \to \phi_{K,\mu}(\xi) = \mathsf{MMD}^2_K(\mu,\xi)$ using a gradient descent algorithm. This gives a family of methods for which performance bounds can be easily established by convexity arguments, arguments that are also applicable to the derivation of performance bounds for the GM and SBQ algorithms.

For any $\xi,\nu \in \mathscr{M}(\mathscr{X})$, the directional derivative of $\phi_{K,\mu}(\cdot)$ at ξ in the direction ν equals

$$F_{\mathsf{MMD}^2_K}(\xi,\nu) = 2\,\left[\int_{\mathscr{X}^2} K_\mu(\mathbf{x},\mathbf{x}')\,\mathrm{d}\nu(\mathbf{x})\,\mathrm{d}\xi(\mathbf{x}') - \mathscr{E}_{K_\mu}(\xi) + \int_{\mathscr{X}} P_{K,\mu}(\mathbf{x})\,\mathrm{d}(\xi-\nu)(\mathbf{x})\right]\,,$$

so that $F_{\mathsf{MMD}^2_K}(\xi,\delta_{\mathbf{x}}) = 2\,[P_{K,\xi}(\mathbf{x}) - P_{K,\mu}(\mathbf{x}) + \int_{\mathscr{X}} P_{K,\mu}(\mathbf{x}')\,\mathrm{d}\xi(\mathbf{x}') - \mathscr{E}_K(\xi)]$. Iterations of the Frank-Wolfe algorithm (Frank and Wolfe, 1956) correspond to $\xi_{k+1} = (1-\alpha_{k+1})\,\xi_k + \alpha_{k+1}\nu_{k+1}$, where $\nu_{k+1} \in \operatorname{Arg\,min}_{\nu\in\mathscr{M}^+_{[1]}(\mathscr{X})} F_{\mathsf{MMD}^2_K}(\xi_k,\nu)$ and $\alpha_{k+1} \in [0,1]$. This gives $\nu_{k+1} = \delta_{\mathbf{x}_{k+1}}$, with

$$\mathbf{x}_{k+1} \quad\in\quad \operatorname{Arg\,min}_{\mathbf{x}\in\mathscr{X}} F_{\mathsf{MMD}^2_K}(\xi_k,\delta_{\mathbf{x}}) = \operatorname{Arg\,min}_{\mathbf{x}\in\mathscr{X}} \left[P_{K,\xi_k}(\mathbf{x}) - P_{K,\mu}(\mathbf{x})\right]\,. \tag{15}$$

When $\alpha_k = 1/k$ for all k, ξ_k remains uniform on its support (unless the same $\mathbf{x}$ is chosen several times); see Wynn (1970) for an early contribution in optimal design of experiments. The method is also called conditional-gradient and corresponds to kernel herding (KH) used in machine learning (Bach et al., 2012). The algorithm selects $\mathbf{x}_1 \in \operatorname{Arg\,max}_{\mathbf{x}\in\mathscr{X}} P_{K,\mu}(\mathbf{x})$ and then

$$\mathbf{x}_{k+1} \in \operatorname{Arg\,min}_{\mathbf{x}\in\mathscr{X}} \sum_{i=1}^{k} K(\mathbf{x}_i,\mathbf{x}) - k\,P_{K,\mu}(\mathbf{x})\,,\ k \geq 1\,. \tag{16}$$

Notice the similarity (but not full agreement) with (14). In particular, (16) can be used with singular kernels, which have an intrinsic repelling property (Pronzato and Zhigljavsky, 2021)

but for which $K(\mathbf{x},\mathbf{x})$ is not defined, whereas (14) cannot. The complexity of KH is $\mathcal{O}(n\,C)$ for n iterations when the selection is among C eligible candidates. In Section 3.2 we consider nonuniform weights, including the case where α_{k+1} is chosen optimally in $\xi_{k+1} = (1-\alpha_{k+1})\,\xi_k + \alpha_{k+1}\delta_{\mathbf{x}_{k+1}}$. In the area of optimal design of experiments, this corresponds to Fedorov's algorithm (1972).

We shall also consider two variants of KH (Section 3.3). First, in a Bayesian integration application, at iteration k of the algorithm we can exploit the support of ξ_k only, and use one of the optimal measures ξ_k^*, $\widehat{\xi}_k$, or $\widetilde{\xi}_k$, with respective weights $\mathbf{w}_k^*$, $\widehat{\mathbf{w}}_k$ and $\widetilde{\mathbf{w}}_k$, for integration; we shall call this variant *Off-Line Weight Optimisation* (OLWO)[3]. Second, we can replace ξ_k by ξ_k^*, $\widehat{\xi}_k$, or $\widetilde{\xi}_k$, *in the algorithm itself* before next iteration; we shall call this variant, closely related to SBQ, *Integrated Weight Optimisation* (IWO)[4].

2.4 Notation

At each iteration, instead of searching $\mathbf{x}$ in the whole set $\mathscr{X}$, we shall use a finite subset $\mathscr{X}_C = \{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(C)}\}$ of candidate points in $\mathscr{X}$ (typically, C points independently sampled from μ; see Section 6). We denote $\mathbb{I}_C = \{1,\ldots,C\}$,

$$\begin{aligned}
\overline{K}_C &= \max_{\mathbf{x}\in\mathscr{X}_C} K(\mathbf{x},\mathbf{x}) \le \overline{K}\,,\\
\overline{K}_{\mu,C} &= \max_{\mathbf{x}\in\mathscr{X}_C} K_\mu(\mathbf{x},\mathbf{x}) = \max_{\mathbf{x}\in\mathscr{X}_C} K(\mathbf{x},\mathbf{x}) - 2\,P_{K,\mu}(\mathbf{x}) + \mathscr{E}_K(\mu)\\
&\le \overline{K}_C + 2\,\overline{K}_C^{1/2}\tau_{1/2}(\mu) + \tau_{1/2}^2(\mu) = \left[\overline{K}_C^{1/2} + \tau_{1/2}(\mu)\right]^2 \qquad (17)
\end{aligned}$$

(and $\overline{K}_{\mu,C} \le \overline{K}_C + \tau_{1/2}^2(\mu)$ when $K \ge 0$). We also denote by $\mathbf{K}_C$ and $\mathbf{K}_{\mu_C}$ the $C\times C$ matrices with i,j elements respectively equal to $K(\mathbf{x}^{(i)},\mathbf{x}^{(j)})$ and $K_\mu(\mathbf{x}^{(i)},\mathbf{x}^{(j)})$, for $\mathbf{x}^{(i)}$, $\mathbf{x}^{(j)}$ in $\mathscr{X}_C$; $\mathbf{e}_j$ is the j-th canonical basis vector of $\mathbb{R}^C$. We assume that μ is not fully supported on $\mathscr{X}_C$, so that $\mathbf{K}_{\mu_C}$ has full rank; see Section 2.3.1 (K_μ is thus s.p.d. on $\mathscr{X}_C$).

For any $\xi \in \mathscr{M}(\mathscr{X})$, any $\mathbf{x}\in\mathscr{X}$ and any $\alpha\in[0,1]$, we define

$$\xi^+(\mathbf{x},\alpha) = (1-\alpha)\,\xi + \alpha\,\delta_{\mathbf{x}}\,, \qquad (18)$$

so that $\xi^+(\mathbf{x},\alpha) \in \mathscr{M}_{[1]}^+(\mathscr{X}_C)$ (respectively, $\mathscr{M}_{[1]}(\mathscr{X}_C)$) when $\mathbf{x}\in\mathscr{X}_C$ and $\xi\in\mathscr{M}_{[1]}^+(\mathscr{X}_C)$ (respectively, $\xi\in\mathscr{M}_{[1]}(\mathscr{X}_C)$).

Any probability measure ξ in $\mathscr{M}_{[1]}^+(\mathscr{X}_C)$, i.e., supported on $\mathscr{X}_C$, can be represented as a vector of weights ω in the probability simplex

$$\mathscr{P}_C = \{\omega\in\mathbb{R}^C : \sum_{j=1}^C \omega_j = 1\,,\ \omega_j \ge 0 \text{ for all } j\}\,.$$

Any measure ξ_n with n support points in $\mathscr{X}_C$ can thus be represented as in (3), with $\mathbf{w}_n$ a n-dimensional vector of weights attached to its support ($\mathbf{w}_n\in\mathscr{P}_n$ when $\xi_n\in\mathscr{M}_{[1]}^+(\mathscr{X}_C)$), and also as a vector $\omega_n\in\mathbb{R}^C$ ($\omega_n\in\mathscr{P}_C$ when $\xi_n\in\mathscr{M}_{[1]}^+(\mathscr{X}_C)$). For any $\mathbf{x}^{(j)}\in\mathscr{X}_C$ and $\xi\in\mathscr{M}(\mathscr{X}_C)$ with weights ω, the vector of weights associated with $\xi^+(\mathbf{x}^{(j)},\alpha)$ equals $\omega^+(\mathbf{x}^{(j)},\alpha) = (1-\alpha)\,\omega + \alpha\mathbf{e}_j$.

We shall denote

[3]It is called Frank-Wolfe Bayesian quadrature in (Briol et al., 2015).

[4]Depending how the 'optimal' measure is constructed, this includes the minimum-norm point (Bach et al., 2012) and fully-corrective Frank-Wolfe (Lacoste-Julien and Jaggi, 2015) algorithms; see Remark 3.

- ξ_*^C the minimum-MMD measure in $\mathscr{M}_{[1]}^+(\mathscr{X}_C)$, with weights $\omega_*^C = \arg\min_{\omega \in \mathscr{P}_C} \omega^\top \mathbf{K}_{\mu C} \omega$, so that (11) gives

$$M_C^2 = \mathsf{MMD}_K^2(\mu, \xi_*^C) = {\omega_*^C}^\top \mathbf{K}_{\mu C} \omega_*^C \,; \tag{19}$$

- $\widehat{\xi}^C$ the minimum-MMD measure in $\mathscr{M}_{[1]}(\mathscr{X}_C)$, with weights $\widehat{\omega}^C$ optimal under the total mass constraint $\mathbf{1}_C^\top \widehat{\omega}^C$ only: $\widehat{\omega}^C$ is given by (5) where $\mathbf{1}_n$, $\mathbf{K}_n$ and $\mathbf{p}_n(\mu)$ are respectively replaced by $\mathbf{1}_C$, $\mathbf{K}_C$ and $\mathbf{p}_C(\mu) = [P_{K,\mu}(\mathbf{x}^{(1)}), \ldots, P_{K,\mu}(\mathbf{x}^{(C)})]^\top$;

- $\widetilde{\xi}^C$ the minimum-MMD unconstrained measure in $\mathscr{M}(\mathscr{X}_C)$, with weights $\widetilde{\omega}^C = \mathbf{K}_C^{-1} \mathbf{p}_C(\mu)$, see (6).

In the rest of the paper we derive finite-sample-size error bounds, i.e., bounds on $\mathsf{MMD}_K^2(\mu, \xi_k)$, for each of the constructions of Section 2.3 and give a numerical illustration in Section 7. Note that we are interested in situations where $k \ll C$. We start with the simplest method, the Frank-Wolfe algorithm, which has already been much studied in the literature (Section 3). The derivation of the upper bound closely follows Clarkson (2010), and those arguments will be central for the derivation of the bounds for GM and SBQ in Sections 4 and 5.

Table 1 summarises our results (error bounds and complexity) for the main algorithms considered. The computational complexity of KH grows like $\mathcal{O}(nC)$ for n iterations; the error bound decreases like $\mathcal{O}((\log n)/n)$ for the standard version with step size $\alpha_k = 1/k$ for all k (which yields uniform weights, Theorem 1) and decreases like $\mathcal{O}(1/n)$ when $\alpha_k = 2/(k+1)$ (Theorem 2) or when α_k is optimised (Theorem 3). The error bounds for the variants OLWO and IWO also decrease like $\mathcal{O}(1/n)$ (Theorem 4) and their computational complexity grows like $\mathcal{O}(n^2 C)$. The same results apply to SBQ (Theorem 8). The numerical experiments in Section 7 indicate that the error bound $\mathcal{O}(1/n)$ for SBQ is pessimistic, in particular for the version where $\mathbf{x}_{k+1}$ is given by (9), some explanations for this pessimism are given in Section 5. GM has the same complexity as KH, with a slightly better error bound for its standard version (Theorem 5) and similar bounds for other versions (Theorems 6 and 7). The case where $\mathscr{X}_C$ is a random set of candidate points, possibly resampled at every iteration, is considered in Section 6 and Appendix C where we show that our results on the decrease of the error bound at finite horizon continue to apply. In some cases, a better bound is obtained when $\widehat{\xi}^C = \xi_*^C$, i.e., when all weights $\widehat{\omega}_i^C$ are nonnegative (which occurs when $\omega_i^* > 0$ for all i). Deriving precise sufficient conditions for this property is a difficult task (as is the question of positivity of quadrature weights in general; see Karvonen et al. (2019)), but it is usually satisfied when the $\mathbf{x}^{(i)}$ in $\mathscr{X}_C$ are independently sampled from μ.

3 Performance analysis of kernel herding and its variants

3.1 Empirical measures

Consider first the case of standard KH, corresponding to Algorithm 1 with $\alpha_k = 1/k$. It selects $\xi_1 = \delta_{\mathbf{x}_1}$, with $\mathbf{x}_1 \in \operatorname{Arg\,max}_{\mathbf{x} \in \mathscr{X}_C} P_{K,\mu}(\mathbf{x})$, and then $\xi_{k+1} = \xi_k^+[\mathbf{x}_{k+1}, 1/(k+1)]$ with $\mathbf{x}_{k+1}$ given by (16) where $\mathscr{X}_C$ is substituted for $\mathscr{X}$. This choice of α_k implies that $\mathbf{w}_k = \mathbf{1}_k/k$ for all k; that is, $\xi_k = \xi_{k,e}$, the empirical measure on $\mathbf{X}_k$. The complexity only grows linearly and is $\mathcal{O}(n\,C)$ for n iterations: the $P_{K,\mu}(\mathbf{x}^{(i)})$ are only computed once for all at the beginning, with complexity $\mathcal{O}(C)$; $S_k(\mathbf{x}^{(i)}) = P_{K,\xi_k}(\mathbf{x}^{(i)})$ is updated at each iteration for each $\mathbf{x}^{(i)}$ in $\mathscr{X}_C$, again

Table 1: Error bound and complexity for n iterations of the KH, GM and SBQ algorithms; $A_C = [\overline{K}_C^{1/2} + \tau_{1/2}(\mu)]^2$, $B_C = 4\,\overline{K}_C$ ($A_C = \overline{K}_C + \tau_{1/2}^2(\mu)$ and $B_C = 2\,\overline{K}_C$ when K is positive), M_C^2 is given by (19).

Method	Algorithm		Error bound	Theorem	Complexity
KH	1	$\alpha_k = 1/k$	$M_C^2 + B_C \frac{2+\log n}{n+1}$	1	$\mathcal{O}(nC)$
	1	$\alpha_k = 2/(k+1)$	$M_C^2 + \frac{4\,B_C}{n+3}$	2	$\mathcal{O}(nC)$
	2	α_k^*	$M_C^2 + \frac{4\,B_C}{n+3}$	3	$\mathcal{O}(nC)$
GM	4	$\alpha_k = 1/k$	$M_C^2 + A_C \frac{1+\log n}{n}$	5	$\mathcal{O}(nC)$
	4	$\alpha_k = 2/(k+1)$	$M_C^2 + \frac{4\,B_C}{n+3}$	6	$\mathcal{O}(nC)$
	5	α_k^*	$M_C^2 + \frac{4\,B_C}{n+3}$	7	$\mathcal{O}(nC)$
SBQ	Eq. (9)		$M_C^2 + \frac{4\,\overline{K}}{n+13/3}$	8	$\mathcal{O}(n^2C)$
	Eq. (13)		$M_C^2 + \frac{4\,B_C}{n+3}$	8	$\mathcal{O}(n^2C)$

Algorithm 1 Kernel herding, predefined step sizes α_k: $\xi_{k+1} = \mathsf{KH}(\xi_k, \alpha_{k+1})$

Require: $\mu \in \mathscr{M}_{[1]}^+(\mathscr{X}) \cap \mathscr{M}_K^{1/2}(\mathscr{X})$, $\mathscr{X}_C \subset \mathscr{X}$, $n \in \mathbb{N}$;
1: set $S_0(\cdot) \equiv 0$ and $\xi_0 = 0$; compute $P_{K,\mu}(\mathbf{x})$ for all $\mathbf{x} \in \mathscr{X}_C$;
2: a sequence $(\alpha_k)_k$ in $[0,1]$ with $\alpha_1 = 1$;
3: $k \leftarrow 1$
4: **while** $k \leq n$ **do**
5: find $\mathbf{x}_k \in \operatorname{Arg\,min}_{\mathbf{x} \in \mathscr{X}_C} S_{k-1}(\mathbf{x}) - P_{K,\mu}(\mathbf{x})$;
6: $S_k(\mathbf{x}) \leftarrow (1-\alpha_k)\, S_{k-1}(\mathbf{x}) + \alpha_k\, K(\mathbf{x}_k, \mathbf{x})$ for all $\mathbf{x} \in \mathscr{X}_C$;
7: $\xi_k \leftarrow (1-\alpha_k)\, \xi_{k-1}(\mathbf{x}) + \alpha_k\, \delta_{\mathbf{x}_k}$;
8: $k \leftarrow k+1$
9: **end while**
10: **return** $\mathbf{X}_n = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, ξ_n.

with complexity $\mathcal{O}(C)$. The finite-sample-size error can be bounded as indicated in Theorem 1. The proof is given in Appendix B.

Theorem 1. *The empirical measure ξ_n generated by Algorithm 1 with $\alpha_k = 1/k$ for all k satisfies*

$$\mathsf{MMD}_K^2(\mu, \xi_n) \leq M_C^2 + B_C\, \frac{2+\log n}{n+1}\,,\ n \geq 1\,, \tag{20}$$

where $B_C = 4\,\overline{K}_C$ ($B_C = 2\,\overline{K}_C$ when K is positive) and M_C^2 is given by (19). *When $\widehat{\xi}^C = \xi_*^C$, ξ_n satisfies*

$$\mathsf{MMD}_K^2(\mu, \xi_n) \leq M_C^2 + \frac{B_C}{n}\,,\ n \geq 1\,. \tag{21}$$

It may happen that the same $\mathbf{x}^{(j)}$ is selected several (possibly consecutive) times at line 5 of Algorithm 1. One may refer to Chen et al. (2018) for the extension to the case where $\mathbf{x}_{k+1}$ is searched within the whole set $\mathscr{X}$ and the selection is suboptimal with some bounded error. Chen et al. (2010) show that the error can decrease as $\mathcal{O}(n^{-2})$ when $\mathcal{H}_K$ is finite-dimensional, but

Bach et al. (2012) indicate that one can only expect the rate $\mathcal{O}(n^{-1})$ in the infinite-dimensional situation; see also Pronzato and Zhigljavsky (2020, Appendix A). In the next section, we show that a better convergence rate than (20), without the log term, can be obtained in general (without the assumption that $\widehat{\xi}^C = \xi_*^C$) when we allow ξ_k to be nonuniform on $\mathbf{X}_k$. The arguments are similar to those used for the proof of Theorem 1: exploiting the convexity of $\mathsf{MMD}_K^2(\mu, \xi_n)$ with respect to the vector of weights ω_n, we obtain a recurrence equation which imposes a particular decrease, see Lemma 2 in Appendix B. The same arguments are used for the other algorithms in the following sections.

3.2 Nonuniform weights

Next theorem shows that for a suitable predefined step-size sequence $(\alpha_k)_k$ in Algorithm 1, the squared MMD decreases as $\mathcal{O}(n^{-1})$. The proof is in Appendix B.

Theorem 2. *The measure ξ_n generated with Algorithm 1 with $\alpha_k = 2/(k+1)$ for all k satisfies*

$$\mathsf{MMD}_K^2(\mu, \xi_n) \leq M_C^2 + \frac{4\, B_C}{n+3}\,, \ n \geq 1\,. \tag{22}$$

Again, it may happen that the same $\mathbf{x}^{(j)}$ is selected several times at line 5 of Algorithm 1; that is, there may exist repetitions in $\mathbf{X}_k$. The weights $\{\mathbf{w}_n\}_i$ that ξ_n allocates to the $\mathbf{x}_i$, $i = 1, \ldots, n$, can be computed explicitly. When $\alpha_k = 2/(k+1)$ for all k, we have $\xi_n = \sum_{i=1}^n 2i/[n(n+1)]\, \delta_{\mathbf{x}_i}$. The distribution is thus far from being uniform, contrary to the case with $\alpha_k = 1/k$; see the right panel of Figure 1. When the condition $\widehat{\xi}^C = \xi_*^C$ is satisfied in Theorem 1, the bound (21) is better than (22) and there is no point in using $\alpha_k = 2/(k+1)$ rather than $\alpha_k = 1/k$. Example 1 will illustrate that the decrease of $\mathsf{MMD}_K(\mu, \xi_k)$ may be worse for nonuniform weights; see Figure 1-left.

As a further attempt to improve performance, we can select $\mathbf{x}_{k+1} \in \operatorname{Arg\,min}_{\mathbf{x} \in \mathscr{X}_C} [P_{K,\xi_k}(\mathbf{x}) - P_{K,\mu}(\mathbf{x})]$ as previously, and then optimise $\mathsf{MMD}_K[\mu, \xi_k^+(\mathbf{x}_{k+1}, \alpha)]$ with respect to α in $[0, 1]$. This function being quadratic in α, the optimal value $\alpha_{k+1}^* = \alpha_{k+1}^*(\mathbf{x}_{k+1})$ can be obtained explicitly; the construction is summarised in Algorithm 2 (see the proof of Theorem 3 in Appendix B for details). Again, the complexity grows linearly with n. The use of the optimal α implies that the same $\mathbf{x}^{(j)}$ cannot be selected two consecutive times at line 4 of Algorithm 2.

Theorem 3. *The measure ξ_n generated with Algorithm 2 satisfies* (22)*; when $\widehat{\xi}^C = \xi_*^C$ it satisfies* (21)*. The optimal α at iteration k is $\alpha_{k+1}^* = \min\{\widehat{\alpha}_{k+1}, 1\}$ with*

$$\widehat{\alpha}_{k+1} = \widehat{\alpha}_{k+1}(\mathbf{x}_{k+1}) = \frac{\sum_{i=1}^k \{\mathbf{w}_k\}_i \, [P_{K,\xi_k}(\mathbf{x}_i) - P_{K,\mu}(\mathbf{x}_i)] + P_{K,\mu}(\mathbf{x}_{k+1}) - P_{K,\xi_k}(\mathbf{x}_{k+1})}{\sum_{i=1}^k \{\mathbf{w}_k\}_i P_{K,\xi_k}(\mathbf{x}_i) - 2\, P_{K,\xi_k}(\mathbf{x}_{k+1}) + K(\mathbf{x}_{k+1}, \mathbf{x}_{k+1})} \tag{23}$$

where $\mathbf{x}_{k+1} \in \operatorname{Arg\,min}_{\mathbf{x} \in \mathscr{X}_C} [P_{K,\xi_n}(\mathbf{x}) - P_{K,\mu}(\mathbf{x})]$. If the algorithm stops at iteration k with $\widehat{\alpha}_{k+1} = 0$, then $\mathsf{MMD}_K(\mu, \xi_k) = \mathsf{MMD}_K(\mu, \xi_*^C)$.

It may seem surprising that the bound obtained with optimal step sizes is not better than when $\alpha_k = 2/(k+1)$ for all k in Algorithm 1, since the decrease of MMD is larger in the former case when starting from the same ξ_k. However, the global decrease over many iterations with the optimal α is not necessarily better than with a predefined step-size sequence; one can refer to Dunn (1980) for a discussion. A numerical comparison in provided in Section 7, showing that Algorithm 2 may perform worse than Algorithm 1; see the left panel of Figure 1.

Algorithm 2 Kernel herding, optimal step sizes: $\xi_{k+1} = \mathsf{KH}(\xi_k, \alpha^*_{k+1})$

Require: $\mu \in \mathscr{M}^+_{[1]}(\mathscr{X}) \cap \mathscr{M}^{1/2}_K(\mathscr{X})$, $\mathscr{X}_C \subset \mathscr{X}$, $n \in \mathbb{N}$;
set $S_0(\cdot) \equiv 0$ and $\xi_0 = 0$; compute $P_{K,\mu}(\mathbf{x})$ for all $\mathbf{x} \in \mathscr{X}_C$;
$k \leftarrow 1$
while $k \leq n$ **do**
  find $\mathbf{x}_k \in \operatorname{Arg\,min}_{\mathbf{x} \in \mathscr{X}_C} S_{k-1}(\mathbf{x}) - P_{K,\mu}(\mathbf{x})$;
  if $k = 1$ **then** set $\alpha^*_k = 1$, $Q_1 = K(\mathbf{x}_1, \mathbf{x}_1)$, $R_1 = P_{K,\mu}(\mathbf{x}_1)$;
  else compute $A_k = Q_{k-1} - R_{k-1} + P_{K,\mu}(\mathbf{x}_k) - S_{k-1}(\mathbf{x}_k)$,
    $B_k = Q_{k-1} - 2\, S_{k-1}(\mathbf{x}_k) + K(\mathbf{x}_k, \mathbf{x}_k)$,
    and $\alpha^*_k = \min\{A_k/B_k, 1\}$
  end if
  if $\alpha^*_k = 0$ **then return** $\mathbf{X}_{k-1} = [\mathbf{x}_1, \ldots, \mathbf{x}_{k-1}]$, ξ_{k-1} and stop;
  end if
  $R_k \leftarrow (1 - \alpha^*_k)\, R_{k-1} + \alpha^*_k\, P_{K,\mu}(\mathbf{x}_k)$;
  $Q_k \leftarrow (1 - \alpha^*_k)^2\, Q_{k-1} + 2\, \alpha^*_k (1 - \alpha^*_k) S_{k-1}(\mathbf{x}_k) + (\alpha^*_k)^2 K(\mathbf{x}_k, \mathbf{x}_k)$;
  $S_k(\mathbf{x}) \leftarrow (1 - \alpha^*_k)\, S_{k-1}(\mathbf{x}) + \alpha^*_k\, K(\mathbf{x}_k, \mathbf{x})$ for all $\mathbf{x} \in \mathscr{X}_C$;
  $\xi_k \leftarrow (1 - \alpha^*_k)\, \xi_{k-1} + \alpha^*_k\, \delta_{\mathbf{x}_k}$;
  $k \leftarrow k + 1$
end while
return $\mathbf{X}_n = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, ξ_n.

Remark 1. *Dunn and Harshbarger (1978) and Dunn (1980) propose other choices of step-size sequences which we do not consider here. We also do not consider Frank-Wolfe algorithm with away steps (Wolfe, 1970; Atwood, 1973)*[5]*, for which* $\xi_{k+1} = \xi_k + \alpha_{k+1}(\xi_k - \delta_{\mathbf{x}_{j_k}})$ *moves away from one of its support points* $\mathbf{x}_{j_k}$*. Here* $\mathbf{x}_{j_k}$ *is taken in* $\operatorname{Arg\,max}_{\mathbf{x} \in \operatorname{supp}(\xi_k)} F_{\mathsf{MMD}^2_K}(\xi_k, \delta_{\mathbf{x}}) = \operatorname{Arg\,max}_{\mathbf{x} \in \mathscr{X}} [P_{K,\xi_k}(\mathbf{x}) - P_{K,\mu}(\mathbf{x})]$ *and* $\alpha_{k+1} \in [0, \xi_k(\mathbf{x}_{j_k})/[1 - \xi_k(\mathbf{x}_{j_k})]$ *to ensure that* $\xi_{k+1}(\mathbf{x}_{j_k}) \geq 0$*; the decision to use an away step instead of* $\xi_{k+1} = \xi^+_k(\mathbf{x}_{k+1}, \alpha)$ *with* $\mathbf{x}_{k+1}$ *given by* (15) *can rely on the comparison between the criterion values obtained, or on the comparison between the absolute values of the directional derivatives* $|F_{\mathsf{MMD}^2_K}(\xi_k, \delta_{\mathbf{x}_{j_k}})|$ *and* $|F_{\mathsf{MMD}^2_K}(\xi_k, \delta_{\mathbf{x}_{k+1}})|$*. Neither do we consider vertex-exchange methods, for which* $\xi_{k+1} = \xi_k + \alpha_{k+1}(\delta_{\mathbf{x}_{k+1}} - \delta_{\mathbf{x}_{j_k}})$ *for* $\alpha_{k+1} \in [0, \xi_k(\mathbf{x}_{j_k})]$*; see for instance Pronzato and Zhigljavsky (2020, Appendix A.3), Pronzato and Pázman (2013, Chap. 9) and the references therein. These methods prove especially efficient for design problems for which the optimal solution is attained on the boundary of* $\mathscr{P}_C$*, with many components equal to zero, in particular due to their ability to reduce the support of the current measure (when* $\alpha_{k+1} = 1$*). The situation is different for the type of problems we have in mind here, and we can only expect a rate of decrease of the finite-sample-size error similar to Algorithm 2. Lacoste-Julien and Jaggi (2015) give a precise analysis of the convergence of these variants of Frank-Wolfe algorithm and prove that they have a global linear convergence rate (contrary to the original Frank-Wolfe algorithm*[6]*). However, the pyramidal width defined in the same paper (eq. (9)) decreases as* $C^{-1/2}$ *and the constant* ρ *in the linear convergence factor* $\exp(-\rho k)$ *decreases as* $1/C$*.* ◁

[5] See also Todd and Yildirim (2007); Ahipaşaoğlu et al. (2008) for a recent use in the minimum-volume ellipsoid problem.

[6] Linear convergence is obtained for the Frank-Wolfe algorithm under the condition that $\widehat{\omega}^C$ is in the interior of $\mathscr{P}_C$; but even in this favourable case the result has no practical interest for large C; see Pronzato and Zhigljavsky (2020, Lemma A4).

Algorithm 3 Kernel herding + IWO (i), (ii) and (iii)

Require: $\mu \in \mathscr{M}_{[1]}^+(\mathscr{X}) \cap \mathscr{M}_K^{1/2}(\mathscr{X})$, $\mathscr{X}_C \subset \mathscr{X}$, $n \in \mathbb{N}$;
set $S_0(\cdot) \equiv 0$ and $\xi_0 = 0$; compute $P_{K,\mu}(\mathbf{x})$ for all $\mathbf{x} \in \mathscr{X}_C$;
$k \leftarrow 1$
while $k \leq n$ **do**
  find $\mathbf{x}_k \in \operatorname{Arg\,min}_{\mathbf{x} \in \mathscr{X}_C} S_{k-1}(\mathbf{x}) - P_{K,\mu}(\mathbf{x})$;
  compute (i) $\mathbf{w}_k = \mathbf{w}_k^*$ (a QP problem), or (ii) $\mathbf{w}_k = \widehat{\mathbf{w}}_k$ (5), or (iii) $\mathbf{w}_k = \widetilde{\mathbf{w}}_k$ (6),
  compute $S_k(\mathbf{x}) = \sum_{i=1}^k \{\mathbf{w}_k\}_i \, K(\mathbf{x}, \mathbf{x}_i)$ for all $\mathbf{x} \in \mathscr{X}_C$;
  $\xi_k = \sum_{i=1}^k \{\mathbf{w}_k\}_i \, \delta_{\mathbf{x}_i}$;
  $k \leftarrow k + 1$
end while
return $\mathbf{X}_n = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, ξ_n.

3.3 KH with off-line and integrated weight optimisation

3.3.1 Off-Line Weight Optimisation (OLWO)

The first KH variant mentioned in Section 2.3.2 (Frank-Wolfe Bayesian quadrature, Briol et al. (2015)) uses the support $\mathbf{X}_k = [\mathbf{x}_1, \ldots, \mathbf{x}_k]$ obtained at iteration k with Algorithm 1 or 2, and constructs an optimal measure ξ_k^*, $\widehat{\xi}_k$, or $\widetilde{\xi}_k$, respectively in $\mathscr{M}_{[1]}^+(\mathbf{X}_k)$, $\mathscr{M}_{[1]}(\mathbf{X}_k)$ or $\mathscr{M}(\mathbf{X}_k)$, with respective weights $\mathbf{w}_k^*$, obtained as solution of a QP problem, $\widehat{\mathbf{w}}_k$ given by (5), and $\widetilde{\mathbf{w}}_k$ given by (6). Let ξ_k be the measure generated by Algorithm 1 or 2, with support $\mathbf{X}_k$; since ξ_k is a probability measure, $\mathsf{MMD}_K^2(\mu, \widetilde{\xi}_k) \leq \mathsf{MMD}_K^2(\mu, \widehat{\xi}_k) \leq \mathsf{MMD}_K^2(\mu, \xi_k^*) \leq \mathsf{MMD}_K^2(\mu, \xi_k)$ for all k, and the bounds of Theorems 1-3 remain valid.

3.3.2 Integrated Weight Optimisation (IWO)

The situation is more complicated for the second variant of Section 2.3.2, where we substitute $\nu_k \in \{\xi_k^*, \widehat{\xi}_k, \widetilde{\xi}_k\}$ for ξ_k *at every iteration* (the case $\nu_k = \xi_k^*$ corresponds to the fully-corrective Frank-Wolfe algorithm, we do not detail the minimum-norm point algorithm, see Remark 3 below). We only consider the situation where the same choice is applied for all iterations and denote respectively by (i), (ii) and (iii) the three cases $\nu_k = \xi_k^*$, $\nu_k = \widehat{\xi}_k$ and $\nu_k = \widetilde{\xi}_k$ for all k. The choice of $\mathbf{x}_{k+1}$ is the same as for KH, but now there is no α_{k+1} to choose. The construction is summarised in Algorithm 3.

Note that $S_k(\cdot) = P_{K,\nu_k}(\cdot)$ can no longer be computed recursively, so that the complexity grows faster than linearly: at iteration k, the complexity of the determination of $\mathbf{w}_k^*$, $\widehat{\mathbf{w}}_k$ or $\widetilde{\mathbf{w}}_k$ is $\mathcal{O}(m(k))$, independently of C (with, in the last two cases, $m(k) = k^3$ in general and $m(k) = k^2$ if rank-one updating is used to compute $\mathbf{K}_k^{-1}$ in (5) and (6); see Remark 4); the complexity of the computation of all $S_k(\mathbf{x}^{(i)})$ is $\mathcal{O}(k\,C)$ and the complexity for n iterations is thus $\mathcal{O}(n^2\,C)$ for $n \ll C$. Kernel herding with IWO satisfies the error bounds in Theorem 4; the proof is in Appendix B.

Theorem 4. *The measure ξ_n generated by Algorithm 3-(i) satisfies* (22)*; when $\widehat{\xi}^C = \xi_*^C$, it satisfies* (21)*. When using Algorithm 3-(ii), ξ_n satisfies* (22)*; when $\widehat{\xi}^C = \xi_*^C$, it satisfies*

$$\mathsf{MMD}_K^2(\mu, \xi_n) \leq M_C^2 + \frac{B_C}{n+2}\,, \; n \geq 2\,, \tag{24}$$

where $B_C = 4\,\overline{K}_C$ *(*$B_C = 2\,\overline{K}_C$ *when* K *is positive) and* M_C^2 *is given by* (19)*. When using Algorithm 3-(iii),* ξ_n *satisfies*

$$\mathsf{MMD}_K^2(\mu,\xi_n) \leq M_C^2 + \frac{4\,\overline{K}}{n + \frac{4\,\overline{K}}{\mathsf{MMD}_K^2(\mu,\xi_1)} - 1} \leq M_C^2 + \frac{4\,\overline{K}}{n+13/3}\,,\ n \geq 1\,. \tag{25}$$

Remark 2. *The measures used in Algorithms 3-(ii) and (iii) are not constrained to belong to* $\mathscr{M}_{[1]}^+(\mathscr{X}_C)$*, so that the algorithm can still progress when* $\mathsf{MMD}_K^2(\mu,\xi_k) \leq \mathsf{MMD}_K^2(\mu,\xi_*^C) = M_C^2$ *(an obvious indication of the pessimism of the bounds in Theorem 4). We show in Appendix B that the following stopping condition can be added after Step 4 of IWO (ii) and (iii), respectively:*

4'-(*ii*): if $S_{k-1}(\mathbf{x}_k) - P_{K,\mu}(\mathbf{x}_k) \geq S_{k-1}(\mathbf{x}_{k-1}) - P_{K,\mu}(\mathbf{x}_{k-1})$ **then return** $\mathbf{X}_{k-1}$*,* ξ_{k-1} *and stop;*

4'-(*iii*): if $S_{k-1}(\mathbf{x}_k) - P_{K,\mu}(\mathbf{x}_k) \geq 0$ **then return** $\mathbf{X}_{k-1}$*,* ξ_{k-1} *and stop;* ◁

Remark 3. *Algorithm 3-(i) corresponds to the fully-corrective Frank-Wolfe algorithm analysed in (Lacoste-Julien and Jaggi, 2015). The Minimum-norm point variant, based on (Wolfe, 1976), uses a sequence of affine projections based on the calculation of a sequence* $\widehat{\omega}_{k_i}$ *restricted to give nonzero weights to subsets* $\mathscr{S}_{k_i}$ *of* $\mathscr{S}_{k_0} = \mathrm{supp}(\xi_k)$ *(at most* k *weights* $\widehat{\omega}_{k_i}$ *need to be calculated); see Algorithm 5 in (Lacoste-Julien and Jaggi, 2015). Since the* $\widehat{\omega}_{k_i}$ *can be obtained explicitly through* (5)*, this construction is simpler than the fully-corrective Frank-Wolfe algorithm. The bounds in Theorem 4 indicated for Algorithm 3-(i) continue to apply, since we still have* $\Delta_C(\xi_{k+1}) \leq (1-\alpha_{k+1})\,\Delta_C(\xi_k) + B_C\,\alpha_{k+1}^2$*,* $k \geq 1$*, for* $\alpha_{k+1} = 2/(k+2)$*, and* $\Delta_C(\xi_{k+1}) \leq (1-2\,\alpha_{k+1})\,\Delta_C(\xi_k) + B_C\,\alpha_{k+1}^2$*,* $k \geq 1$*, for* $\alpha_{k+1} = 1/(k+1)$ *when* $\widehat{\xi}^C = \xi_*^C$*; see the proofs of Theorems 1 and 4.* ◁

Remark 4. *OLWO and IWO require the repeated computation of optimal weights* $\widehat{\mathbf{w}}_n$ *or* $\widetilde{\mathbf{w}}_n$*, respectively given by* (5) *and* (6)*, for which it is advantageous to use the block matrix inversion* (7)*. Rank-one Cholesky updates can be used too; the details are omitted. Eq.* (7) *also gives*

$$\begin{aligned}
\mathbf{k}_{n+1}^\top(\mathbf{x})\mathbf{K}_{n+1}^{-1}\mathbf{k}_{n+1}(\mathbf{x}) &= \mathbf{k}_n^\top(\mathbf{x})\mathbf{K}_n^{-1}\mathbf{k}_n(\mathbf{x}) + \beta_{n+1}\,[(\mathbf{u}_{n+1}^\top,\ -1)\mathbf{k}_{n+1}(\mathbf{x})]^2\,,\\
\mathbf{1}_{n+1}^\top\mathbf{K}_{n+1}^{-1}\mathbf{k}_{n+1}(\mathbf{x}) &= \mathbf{1}_n^\top\mathbf{K}_n^{-1}\mathbf{k}_n(\mathbf{x}) + \beta_{n+1}\,[(\mathbf{u}_{n+1}^\top,\ -1)\mathbf{1}_{n+1}][(\mathbf{u}_{n+1}^\top,\ -1)\mathbf{k}_{n+1}(\mathbf{x})]
\end{aligned}$$

for all $\mathbf{x}$*, so that matrix-vector multiplications can be avoided in SBQ when computing the denominators on the right-hand side of* (9) *and* (13) *by using recursive calculation.* ◁

Remark 5. *The numerical experiments of Section 7 show that the bound* (25) *for Algorithm 3-(iii) becomes very loose when* k *increases. The arguments used in the proof of Theorem 4 suggest two sources of pessimism. First, we substitute the inequality* (51) *for the convexity bound* (50) *(this approximation is used for all the methods considered). Second, we ignore the decrease of* $K(\mathbf{x}_{k+1},\mathbf{x}_{k+1}) - \mathbf{k}_k^\top(\mathbf{x}_{k+1})\mathbf{K}_k^{-1}\mathbf{k}_k(\mathbf{x}_{k+1})$ *as* k *increases in the denominator on the right-hand side of* (8)*, and simply bound it by* $\overline{K}$*. In the algorithm defined by* (27)*, the denominator is constant, so that the pessimism of the error bound is mainly due to the substitution of* (51) *for* (50)*; this effect is illustrated on Figure 2-left. Although this indicates that there still exists room for improvement, the derivation of better bounds seems difficult. Similar statements can be made for Algorithm 3-(ii), for which numerical experiments show that it performs similarly to Algorithm 1 for moderate* k*, but tends to converge faster for large* k *(see Figure 2-left and Figure 6-left).* ◁

Algorithm 4 Greedy MMD minimisation, predefined step sizes α_k: $\xi_{k+1} = \mathsf{GM}(\xi_k, \alpha_{k+1})$

Require: $\mu \in \mathscr{M}_{[1]}^+(\mathscr{X}) \cap \mathscr{M}_K^{1/2}(\mathscr{X})$, $\mathscr{X}_C \subset \mathscr{X}$, $n \in \mathbb{N}$;
set $S_0(\cdot) \equiv 0$ and $\xi_0 = 0$; compute $K(\mathbf{x}, \mathbf{x})$ and $P_{K,\mu}(\mathbf{x})$ for all $\mathbf{x} \in \mathscr{X}_C$;
a sequence $(\alpha_k)_k$ in $[0, 1]$ with $\alpha_1 = 1$;
$k \leftarrow 1$
while $k \leq n$ **do**
  find $\mathbf{x}_k \in \operatorname{Arg\,min}_{\mathbf{x} \in \mathscr{X}_C} 2(1 - \alpha_k)\, S_{k-1}(\mathbf{x}) + \alpha_k\, K(\mathbf{x}, \mathbf{x}) - 2\, P_{K,\mu}(\mathbf{x})$;
  $S_k(\mathbf{x}) \leftarrow (1 - \alpha_k)\, S_{k-1}(\mathbf{x}) + \alpha_k\, K(\mathbf{x}_k, \mathbf{x})$ for all $\mathbf{x} \in \mathscr{X}_C$;
  $\xi_k \leftarrow (1 - \alpha_k)\, \xi_{k-1}(\mathbf{x}) + \alpha_k\, \delta_{\mathbf{x}_k}$;
  $k \leftarrow k + 1$
end while
return $\mathbf{X}_n = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, ξ_n.

4 Performance analysis of Greedy MMD Minimisation (GM)

4.1 Empirical measures

GM with empirical measures corresponds to Algorithm 4 with $\alpha_k = 1/k$ for all k; it selects $\mathbf{x}_1 \in \operatorname{Arg\,min}_{\mathbf{x} \in \mathscr{X}_C} K(\mathbf{x}, \mathbf{x}) - 2\, P_{K,\mu}(\mathbf{x})$ and then chooses $\mathbf{x}_{n+1}$ according to (14) with $\mathscr{X}_C$ substituted for $\mathscr{X}$. It corresponds to Algorithm 1 in (Teymur et al., 2021), where the authors derive a finite-sample-size error bound using the RKHS framework. Taking advantage of the finiteness of the candidate set $\mathscr{X}_C$, we provide a simpler proof using only linear (finite-dimensional) algebra; see Appendix B. Notice that the bound is smaller than for KH in Theorem 1. The complexity of Algorithm 4 is $\mathcal{O}(n\,C)$ for n iterations: the $K(\mathbf{x}^{(i)}, \mathbf{x}^{(i)})$ and $P_{K,\mu}(\mathbf{x}^{(i)})$ are only computed once for all at the beginning, with complexity $\mathcal{O}(C)$, the $S_k(\mathbf{x}^{(i)})$ are updated at each iteration for all $\mathbf{x}^{(i)} \in \mathscr{X}_C$, again with complexity $\mathcal{O}(C)$.

Theorem 5. *The measure ξ_n generated by Algorithm 4 with $\alpha_k = 1/k$ for all k satisfies*

$$\mathsf{MMD}_K^2(\mu, \xi_n) \leq M_C^2 + A_C\, \frac{1 + \log n}{n}\,, \quad n \geq 1\,, \tag{26}$$

where M_C^2 is given by (19) *and $A_C = [\overline{K}_C^{1/2} + \tau_{1/2}(\mu)]^2$ ($A_C = \overline{K}_C + \tau_{1/2}^2(\mu)$ when K is positive).*

4.2 Nonuniform weights

Consider now the case of general discrete measures ξ_n in $\mathscr{M}_{[1]}^+(\mathscr{X}_C)$, see (3). We show that allowing nonuniform weights in GM yields a faster decrease of $\mathsf{MMD}_K(\mu, \xi_n)$. As for KH, we consider iterations of the form $\xi_{k+1} = \xi_k^+(\mathbf{x}_{k+1}, \alpha_{k+1})$, $k \geq 1$, for some $\alpha_{k+1} \in [0, 1]$ and $\mathbf{x}_{k+1} \in \mathscr{X}_C$, where $\xi_k^+(\mathbf{x}, \alpha)$ is defined by (18). We first consider the same choice $\alpha_k = 2/(k+1)$ as in Section 3.2. The proof of Theorem 6 is in Appendix B.

Theorem 6. *The measure ξ_n generated by Algorithm 4 with $\alpha_k = 2/(k+1)$ for all k satisfies* (22)*. When $\widehat{\xi}^C = \xi_*^C$, Algorithm 4 with $\alpha_k = 1/k$ for all k yields* (21)*.*

Remark 6. *As for Algorithm 1, when $\alpha_k = 2/(k+1)$ in Algorithm 4 the measure ξ_n is not uniform on its support $\mathbf{X}_n$. It is uniform when $\alpha_k = 1/k$ for all k, but the arguments used in the proof of Theorem 6 only give* (20)*, which is worse than* (26) *obtained by Teymur et al. (2021).* ◁

Algorithm 5 Greedy MMD minimisation, optimal step sizes: $\xi_{k+1}\mathsf{GM}(\xi_k, \alpha^*_{k+1})$

Require: $\mu \in \mathscr{M}^+_{[1]}(\mathscr{X}) \cap \mathscr{M}^{1/2}_K(\mathscr{X})$, $\mathscr{X}_C \subset \mathscr{X}$, $n \in \mathbb{N}$;
compute $K(\mathbf{x}, \mathbf{x})$ and $P_{K,\mu}(\mathbf{x})$ for all $\mathbf{x} \in \mathscr{X}_C$;
set $S_0(\cdot) \equiv 0$, $Q_0 = R_0 = 0$, $\alpha_1(\cdot) \equiv 1$ and $\xi_0 = 0$;
set $A_0(\mathbf{x}) = P_{K,\mu}(\mathbf{x})$, $B_0(\mathbf{x}) = K(\mathbf{x}, \mathbf{x})$ for all $\mathbf{x} \in \mathscr{X}_C$;
$k \leftarrow 1$
while $k \leq n$ **do**
    find $\mathbf{x}_k \in \operatorname{Arg\,min}_{\mathbf{x} \in \mathscr{X}_C} \alpha^2(\mathbf{x}) B_{k-1}(\mathbf{x}) - 2\,\alpha(\mathbf{x}) A_{k-1}(\mathbf{x})$;
    $\alpha_k \leftarrow \alpha(\mathbf{x}_k)$
    $R_k \leftarrow (1 - \alpha_k)\, R_{k-1} + \alpha_k\, P_{K,\mu}(\mathbf{x}_k)$;
    $Q_k \leftarrow (1 - \alpha_k)^2\, Q_{k-1} + 2\,\alpha_k (1 - \alpha_k) S_{k-1}(\mathbf{x}_k) + \alpha_k^2 K(\mathbf{x}_k, \mathbf{x}_k)$;
    $S_k(\mathbf{x}) \leftarrow (1 - \alpha_k)\, S_{k-1}(\mathbf{x}) + \alpha_k\, K(\mathbf{x}_k, \mathbf{x})$ for all $\mathbf{x} \in \mathscr{X}_C$;
    $\xi_k \leftarrow (1 - \alpha_k)\, \xi_k + \alpha_k\, \delta_{\mathbf{x}_k}$;
    compute $A_k(\mathbf{x}) = Q_k - R_k + P_{K,\mu}(\mathbf{x}) - S_k(\mathbf{x})$, $B_k(\mathbf{x}) = Q_k - 2\, S_k(\mathbf{x}) + K(\mathbf{x}, \mathbf{x})$
        and $\alpha(\mathbf{x}) = \max\{0, \min\{A(\mathbf{x})/B(\mathbf{x}), 1\}\}$ for all $\mathbf{x} \in \mathscr{X}_C$;
    if all $\alpha(\mathbf{x})$ equal $= 0$ **then return** $\mathbf{X}_k = [\mathbf{x}_1, \ldots, \mathbf{x}_k]$, ξ_k and stop;
    end if
    $k \leftarrow k + 1$
end while
return $\mathbf{X}_n = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, ξ_n.

Consider now GM with optimal step size, which selects α_{k+1} and $\mathbf{x}_{k+1}$ optimally at each iteration: $[\mathbf{x}_{k+1}, \alpha_{k+1}] \in \operatorname{Arg\,min}_{\mathbf{x} \in \mathscr{X}_C,\, \alpha \in [0,1]} \mathsf{MMD}_K[\mu, \xi_k^+(\mathbf{x}, \alpha)]$, with $\xi_1 = \delta_{\mathbf{x}_1}$ and $\mathbf{x}_1 \in \operatorname{Arg\,min}_{\mathbf{x} \in \mathscr{X}_C} K(\mathbf{x}, \mathbf{x}) - 2\, P_{K,\mu}(\mathbf{x})$. As in Algorithm 2, the optimal value of $\alpha_{k+1} = \alpha(\mathbf{x}_{k+1})$ is obtained explicitly, see the proof of Theorem 7 in Appendix B for details. The complexity is again $\mathcal{O}(n\,C)$ for n iterations (it is larger than for Algorithm 2 as $\alpha(\mathbf{x})$ must be calculated for all $\mathbf{x} \in \mathscr{X}_C$).

Theorem 7. *The measure ξ_n generated with Algorithm 5 satisfies* (22)*. When $\widehat{\xi}^C = \xi_*^C$, it satisfies* (21)*.*

Similarly to Theorem 3, one might think that bounds obtained with predefined step sizes should be overly loose concerning an algorithm for which α_k is optimised at every iteration. However, the observed behaviour is often similar to that of Algorithm 4, if not worse, see the examples in Section 7, indicating that optimal but myopic steps are not necessarily preferable to myopic, non-optimised but suitably chosen steps; see, e.g., Zhigljavsky et al. (2012) for an illustration with the steepest descent algorithm.

Remark 7. *As for KH, one may also consider OLWO and IWO variants of GM; see Section 3.3. The OLWO variant does not raise any particular difficulty as it runs in parallel and does not affect the algorithm: like in Section 3.3.1 for KH, OLWO can only improve performance. The situation is different for IWO: the fact that the next point $\mathbf{x}_{k+1}$ and the step size α_{k+1} must be selected simultaneously render its use less adapted than with KH, which maximises the right-hand side of* (9) *and* (13)*, see the proof of Theorem 4.* ◁

5 Performance analysis of SBQ

We suppose again that the successive support points are searched within a finite candidate set $\mathscr{X}_C$, and consider the two versions of SBQ presented in Section 2.3.1 with $\mathscr{X}_C$ substituted for $\mathscr{X}$. We do not detail the algorithm which simply implements (9) or (13)—with $\mathbf{K}_k^{-1}$ calculated recursively as indicated in Remark 4. The numerical experiments of Section 7 show that the two versions behave similarly to Algorithms 3-(*iii*) and (*ii*), respectively, but have a slightly higher computational cost.

We also consider a version of SBQ where all previous weights $\{\widetilde{\mathbf{w}}_k\}_i$ are kept fixed, $i = 1, \ldots, k$, and only the next one is optimised (without constraint) when choosing $\mathbf{x}_{k+1}$. Since $\mathsf{MMD}_K^2(\mu, \xi_k + w\,\delta_{\mathbf{x}}) = \mathsf{MMD}_K^2(\mu, \xi_k) + w^2\,K(\mathbf{x},\mathbf{x}) + 2w\,[P_{K,\xi_k}(\mathbf{x}) - P_{K,\mu}(\mathbf{x})]$, the optimal w is $w^*(\mathbf{x}) = [P_{K,\mu}(\mathbf{x}) - P_{K,\widetilde{\xi}_k}(\mathbf{x})]/K(\mathbf{x},\mathbf{x})$. This algorithm selects $\mathbf{x}_1 \in \mathrm{Arg}\max_{\mathbf{x}\in\mathscr{X}_C} P_{K,\mu}^2(\mathbf{x})/K(\mathbf{x},\mathbf{x})$ and then uses

$$\mathbf{x}_{k+1} \in \mathrm{Arg}\max_{\mathbf{x}\in\mathscr{X}_C} \frac{\left[P_{K,\widetilde{\xi}_k}(\mathbf{x}) - P_{K,\mu}(\mathbf{x})\right]^2}{K(\mathbf{x},\mathbf{x})}\,,\ \xi_{k+1} = \xi_k + w^*(\mathbf{x}_{k+1})\delta_{\mathbf{x}_{k+1}}\,,\ k \geq 1\,. \tag{27}$$

When $K(\mathbf{x},\mathbf{x})$ is a constant, the choice of $\mathbf{x}_{k+1}$ is similar to that of KH, see (15). It corresponds to a Coordinate-Descent (CD) algorithm (see, e.g., Wright (2015)) operating on the weights $\omega = (\omega_1, \ldots, \omega_C) \in \mathbb{R}^C$; see Section 2.4. Performance bounds for these three versions of SBQ are given in Theorem 8.

Theorem 8. *Suppose that $\mathscr{X}_C$ is substituted for $\mathscr{X}$. Then, $\mathsf{MMD}_K^2(\mu, \xi_n)$ satisfies the same bounds as those indicated in Theorem 4 for Algorithm 3-(ii) when using* (13)*, and satisfies* (25) *when using* (9)*, or when using* (27) *if $K(\mathbf{x},\mathbf{x})$ is a constant.*

Although our numerical experiments indicate that (27) is not competitive compared to (9), the analysis of its finite sample error helps understanding the pessimism of the error bound derived for version (9) of SBQ; see Remark 5 and the proof of Theorem 8.

6 Random candidate sets

The extension of the results in previous sections to the case where $\mathscr{X}_C$ corresponds to C points independently sampled from μ is fairly simple; see Teymur et al. (2021). For instance, (22) becomes

$$\mathsf{E}_\mu\{\mathsf{MMD}_K^2(\mu, \xi_n)\} \leq \mathsf{E}_\mu\{M_C^2\} + \frac{4\,\mathsf{E}_\mu\{B_C\}}{n+3}\,,\ n \geq 1\,.$$

We thus only have to bound the expected values of the constants A_C, B_C and M_C^2 that intervene in the various bounds that have been presented. Their values are given in the following lemma, the proof is in Appendix B.

Lemma 1. *Suppose $\mu \in \mathscr{M}_K^1(\mathscr{X})$ and that the C points in $\mathscr{X}_C$ are independently sampled from μ, then*

$$\begin{aligned}
\mathsf{E}_\mu\{A_C\} &\leq A(\mu) = [\overline{K}^{1/2} + \tau_{1/2}(\mu)]^2 \ (A(\mu) = \overline{K} + \tau_{1/2}^2(\mu) \text{ when } K \text{ is positive}),\\
\mathsf{E}_\mu\{B_C\} &\leq B = 4\,\overline{K} \ (B = 2\,\overline{K} \text{ when } K \text{ is positive}),\\
\mathsf{E}_\mu\{M_C^2\} &\leq M^2(\mu)/C = [\tau_1(\mu) - \mathscr{E}_K(\mu)]/C\,,
\end{aligned}$$

where M_C^2 *is given by* (19), $A_C = [\overline{K}_C^{1/2} + \tau_{1/2}(\mu)]^2$ *and* $B_C = 4\,\overline{K}_C$ *(*$A_C = \overline{K}_C + \tau_{1/2}^2(\mu)$ *and* $B_C = 2\,\overline{K}_C$ *when* K *is positive).*

Teymur et al. (2021) derive a bound similar to (26) (with A_C and M_C^2 replaced by $\mathsf{E}_\mu\{A_C\}$ and $\mathsf{E}_\mu\{M_C^2\}$) for Algorithm 4 with $\alpha_k = 1/k$ for all k in the situation where a different sample $\mathscr{X}_C[k]$ of C random points is used at each iteration; see also Chen et al. (2019). The extension to this situation of the approach used in previous sections does not seem straightforward as the probability simplex $\mathscr{P}_C$ and matrices $\mathbf{K}_C$ and $\mathbf{K}_{\mu C}$ refer to a fixed set $\mathscr{X}_C$. In Appendix C we provide arguments explaining how our results extend to the case where $\mathscr{X}_C = \mathscr{X}_C[k]$ depends on k: basically, similar bounds continue to hold provided we consider the expectation of $\mathsf{MMD}_K^2(\mu, \xi_n)$ and bound the expected values of the constants involved as in Lemma 1. Note that changing the candidate set at every iteration implies that we need to calculate $K(\mathbf{x}_i, \mathbf{x})$ for all $\mathbf{x}_i \in \mathrm{supp}(\xi_k)$ and all $\mathbf{x} \in \mathscr{X}_C[k]$, and to recalculate $P_{K,\mu}(\mathbf{x})$ (Algorithms 1 and 2), or $P_{K,\mu}(\mathbf{x})$ and $K(\mathbf{x}, \mathbf{x})$ (Algorithms 4 and 5), for all $\mathbf{x} \in \mathscr{X}_C[k]$ at every iteration, with a computational cost thus growing as $\mathcal{O}(k^2\,C)$.

We conclude this section by recalling a result on the MMD of the empirical measure $\xi_{n,e}$ of a random n-point sample from μ; see Mak and Joseph (2018, Lemma 2). The proof is given in Appendix B.

Theorem 9. *When* $\mathbf{x}_1, \ldots, \mathbf{x}_n$ *are independently sampled from* μ*, then* $n\,\mathsf{MMD}_K^2(\mu, \xi_{n,e}) \xrightarrow{\mathrm{d}} Z = \sum_{i=1}^\infty \lambda_i \chi_{1i}^2$*, where the* λ_i *are the eigenvalues of the operator* T_{K_μ} *on* $L_2(\mathscr{X}, \mu)$ *defined by* $T_{K_\mu} f(\mathbf{x}) = \int_{\mathscr{X}} K_\mu(\mathbf{x}, \mathbf{x}') f(\mathbf{x}')\,\mathrm{d}\mu(\mathbf{x}')$*,* $f \in L_2(\mathscr{X}, \mu)$*,* $\mathbf{x} \in \mathscr{X}$*, and the* χ_{1i}^2 *are independent* χ_1^2 *random variables.*

From Lemma 1 and Theorem 9 we have in particular $\mathsf{E}_\mu\{\mathsf{MMD}_K^2(\mu, \xi_{n,e})\} = M^2(\mu)/n = [\tau_1(\mu) - \mathscr{E}_K(\mu)]/n$ and $n^2\,\mathsf{var}_\mu\{\mathsf{MMD}_K^2(\mu, \xi_{n,e})\} \to 2\sum_{i=1}^\infty \lambda_i^2$ as $n \to \infty$. Although the bounds obtained in previous sections suggest that the measures obtained with the algorithms that have been considered do not perform necessarily better (asymptotically) than i.i.d. samples from μ, the examples in the next section demonstrate the interest of using KH, GM or SBQ.

7 Numerical study

7.1 Example 1: space-filling design

For illustration purpose we only consider the case $d = 2$ and take μ uniform on $\mathscr{X} = [0, 1]^2$; $\mathscr{X}_C$ corresponds to the first $2^{17} = 131\,072$ points of a scrambled Sobol' sequence in $\mathscr{X}$. K is a separable kernel given by the product of uni-dimensional Matérn 3/2 covariance functions, that is, $K(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^d K_{3/2,\theta}(x_i, x_i')$ with

$$K_{3/2,\theta}(x, x') = (1 + \sqrt{3}\theta\,|x - x'|)\,\exp(-\sqrt{3}\theta\,|x - x'|)\,.$$

We have $\mathscr{E}_K(\mu) = \prod_{i=1}^d \mathscr{E}_{K_{3/2,\theta}}(\mu_1)$ and $P_{K,\mu}(\mathbf{x}) = \prod_{i=1}^d P_{K_{3/2,\theta},\mu_1}(x_i)$ with μ_1 the uniform measure on $[0, 1]$, and $\mathscr{E}_{K_{3/2,\theta}}(\mu_1)$ and $P_{K_{3/2,\theta},\mu_1}(x)$ can be computed explicitly; see, e.g., Pronzato and Zhigljavsky (2020, Table 3.1). Examples of space-filling design based on MMD-minimisation with $d = 10$ and recommendations for the choice of θ are given in the same paper. We use $\theta = 10$ throughout the example.

The left panel of Figure 1 shows the evolution of $\mathsf{MMD}_K(\mu, \xi_n)$ as a function of n (log-log plot) when ξ_n is generated with Algorithm 1 with $\alpha_k = 1/k$ and $\alpha_k = 2/(k+1)$, or with Algorithm 2. The bound (22) is also shown (M_C^2 is negligible). Although Algorithm 2 uses optimal step-sizes, its performance is the worst for large n and is never better than that of Algorithm 1 with $\alpha_k = 1/k$ (note that the rate of decrease of $\mathsf{MMD}_K(\mu, \xi_n)$ for Algorithm 2 closely follows $\mathcal{O}(1/n)$ when $n \gtrsim 100$). Although the bound (22) of Theorem 2 is better than (20) of Theorem 1, $\alpha_k = 1/k$ yields better performance than $\alpha_k = 2/(k+1)$ all along the sequence. This suggests that there is little interest in using more sophisticated versions of KH than Algorithm 1 with $\alpha_k = 1/k$. We also computed the MMD for empirical measures associated with random designs. The average and 2σ intervals obtained for 100 repetitions are presented, showing a decrease that closely follows $\mathcal{O}(1/n)$, as predicted by Theorem 9. The evolution of $\mathsf{MMD}_K(\mu, \xi_n)$ obtained for ξ_n generated with Algorithm 4 with $\alpha_k = 1/k$ and $\alpha_k = 2/(k+1)$ (respectively, with Algorithm 5) is visually hardly distinguishable from that obtained with Algorithm 1 with $\alpha_k = 1/k$ and $\alpha_k = 2/(k+1)$ (respectively, with Algorithm 2) and is not presented.

The right panel of Figure 1 shows the strong non-uniformity of the weights $\{\mathbf{w}_{1\,000}\}_i$, $i = 1, \ldots, 1\,000$, associated with the measures $\xi_{1\,000}$ generated by Algorithm 1 with $\alpha_k = 2/(k+1)$ and by Algorithm 2 (note that most recent points are overweighed for the former and down-weighed for the latter).

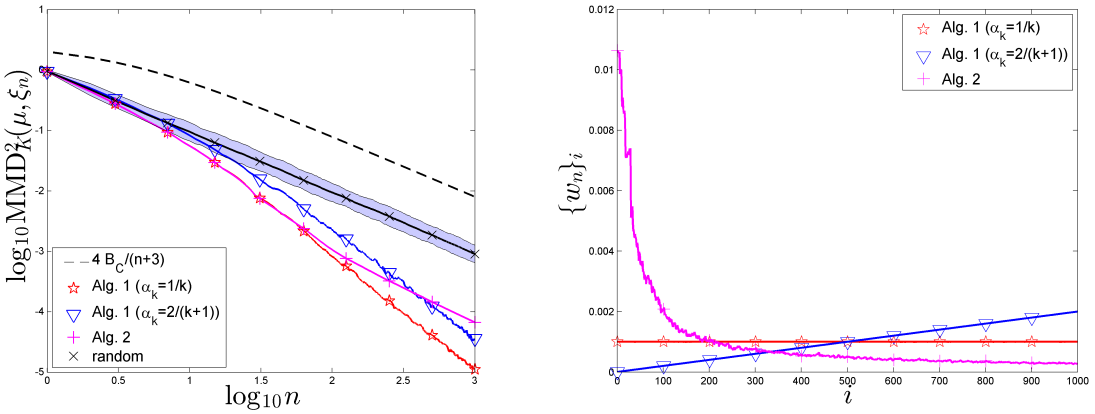

Figure 1: Left: upper bound (22) and $\mathsf{MMD}_K(\mu, \xi_n)$ for ξ_n generated with Algorithm 1 with $\alpha_k = 1/k$ and $\alpha_k = 2/(k+1)$, with Algorithm 2 and for the empirical measure $\xi_n = \xi_{n,e}$ of random n-point designs (mean value $\pm$ 2σ over 100 repetitions), $n = 1, \ldots, 1\,000$. Right: weights $\{\mathbf{w}_{1\,000}\}_i$ of $\xi_{1\,000}$.

We consider now the variants (ii) and (iii) of IWO in Algorithm 3. Unsurprisingly, $\mathsf{MMD}_K(\mu, \nu_n)$ is smaller for $\nu_n = \widetilde{\xi}_n$ of variant (iii) than for $\nu_n = \widehat{\xi}_n$ of variant (ii) since the weights are unconstrained in the former case, see the left panel of Figure 2. The two variants perform similarly for large n, however. The bound (25) for Algorithm 3-(iii) is accurate for small n but very pessimistic for large n; see Remark 5. Algorithm 3-(ii) performs as Algorithm 1 with $\alpha_k = 1/k$ for small n ($n \lesssim 30$) but performs significantly better for larger n. The performances are quasi identical when using OLWO of Section 3.3.1 (Frank-Wolfe Bayesian quadrature, not shown). When we stop Algorithm 1 (with $\alpha_k = 1/k$) at $n = 200$, all weights $\{\widehat{\mathbf{w}}_n\}_i$ and $\{\widetilde{\mathbf{w}}_n\}_i$, $i = 1, \ldots, n$, are positive. The weights are positive too for Algorithm 3-(ii) and (iii), so that variant (ii)

coincides with Algorithm 3-(i), the fully-corrective Frank-Wolfe algorithm (and also with the minimum-norm point algorithm, see Remark 3). The evolution of $\mathsf{MMD}_K(\mu, \xi_n)$ for ξ_n obtained with the version (13) of SBQ (with weights whose sum equals one) is indistinguishable from that obtained with Algorithm 3-(ii). The behaviour of the version (9) of SBQ (with unconstrained weights) is similar to that of Algorithm 3-(iii) and is rather typical, see for instance Briol et al. (2015); Huszár and Duvenaud (2012); see also Figure 6-left. The Coordinate-Descent variant (27) of SBQ, denoted SBQ-CD, is clearly not competitive compared to the other algorithms considered.

Computational times[7] are shown on the right panel of Figure 2 for Algorithm 1 with $\alpha_k = 1/k$ and $\alpha_k = 2/(k+1)$, for Algorithm 2, and for Algorithm 3-(ii) and (iii)[8]. The choice of the sequence $(\alpha_k)_k$ has no influence on the computational time of Algorithm 1; Algorithm 2 is slightly more demanding, but its computational time still grows linearly with n; the better performance of IWO shown on the left panel comes with a significant increase of computational cost (which is similar for OLWO).

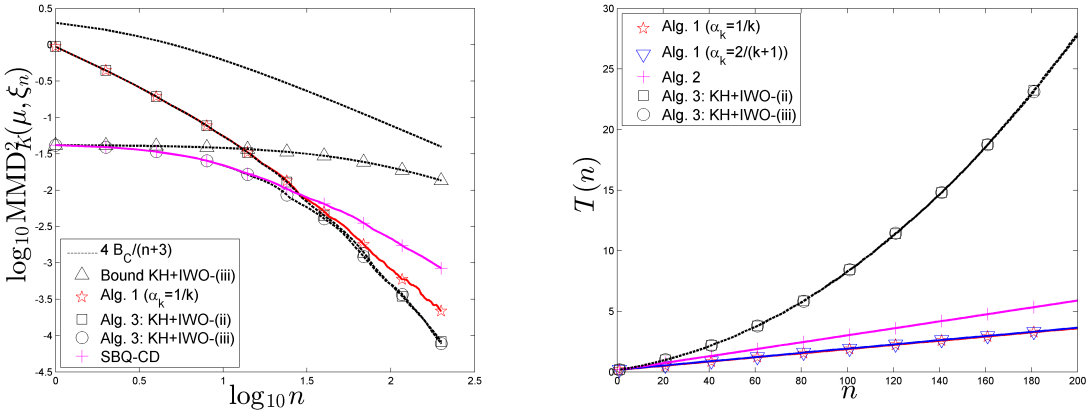

Figure 2: Left: upper bounds (22) and (25), and $\mathsf{MMD}_K(\mu, \xi_n)$ for ξ_n generated with Algorithm 1 with $\alpha_k = 1/k$, with Algorithm 3-(ii) and (iii), and with version (27) of SBQ, $n = 1, \ldots, 200$. Right: computational time $T(n)$ (in s) of ξ_n for Algorithm 1 with $\alpha_k = 1/k$ and $\alpha_k = 2/(k+1)$, for Algorithm 2 and for Algorithm 3-(ii) and (iii), $n = 1, \ldots, 200$.

Computational times for Algorithm 4 with $\alpha_k = 1/k$, Algorithm 5 and the two versions (9) and (13) of SBQ are shown on the left panel of Figure 3: Algorithm 4 is as fast as Algorithm 1; Algorithm 5 is slightly slower than Algorithm 3. The two versions of SBQ are slightly slower than Algorithm 3-(ii) and (iii) for similar performance.

MMD minimisation with μ uniform on $\mathscr{X}$ is an efficient method to construct nested space-filling designs; this is one of the main motivations in (Pronzato and Zhigljavsky, 2020). The right panel of Figure 3 shows the 25-point design corresponding to the support of the measure ξ_n generated with Algorithm 4 with $\alpha_k = 1/k$, with a covering radius[9] $\mathsf{CR}(\mathbf{X}_{25}) \simeq 0.1625$.

[7] All calculations are made with Matlab, on a PC with a clock speed of 1.5 GHz and 16 GB RAM.

[8] All computational times start with a positive value at $n = 0$ since we account for the calculation of $P_{K,\mu}(\mathbf{x})$ for all $\mathbf{x} \in \mathscr{X}_C$. The faster running time than the theoretical complexity estimates given in the paper can be explained by the internal vectorisation of operations in Matlab.

[9] The covering radius of a design $\mathbf{X}_n$ is defined by $\mathsf{CR}(\mathbf{X}_n) = \max_{\mathbf{x}\in\mathscr{X}} \min_{\mathbf{x}_i \in \mathbf{X}_n} \|\mathbf{x} - \mathbf{x}_i\|$; a small value of

Algorithm 1 with $\alpha_k = 1/k$ yields a very similar design, but with a different ordering of points and a slightly larger covering radius $\mathsf{CR}(\mathbf{X}_{25}) \simeq 0.1685$. When using Algorithm 3-(*ii*) (respectively, Algorithm 3-(*iii*)), the support of ξ_{25} has a covering radius $\mathsf{CR}(\mathbf{X}_{25}) \simeq 0.1677$ (respectively, $\mathsf{CR}(\mathbf{X}_{25}) \simeq 0.2024$). This illustrates the fact that a smaller MMD is not necessarily synonym to better space-filling properties: the optimal weighting of a given design improves its MMD, but space-filling performance, measured for instance by the covering radius, is unweighed. In fact, when allocating uniform weights to the support of ξ_n generated with Algorithm 3-(*iii*), the MMD obtained is similar to that shown on the left panel of Figure 2 for Algorithm 1 with $\alpha_k = 1/k$ (red ★), thus much worse than for the original ξ_n (black ∘).

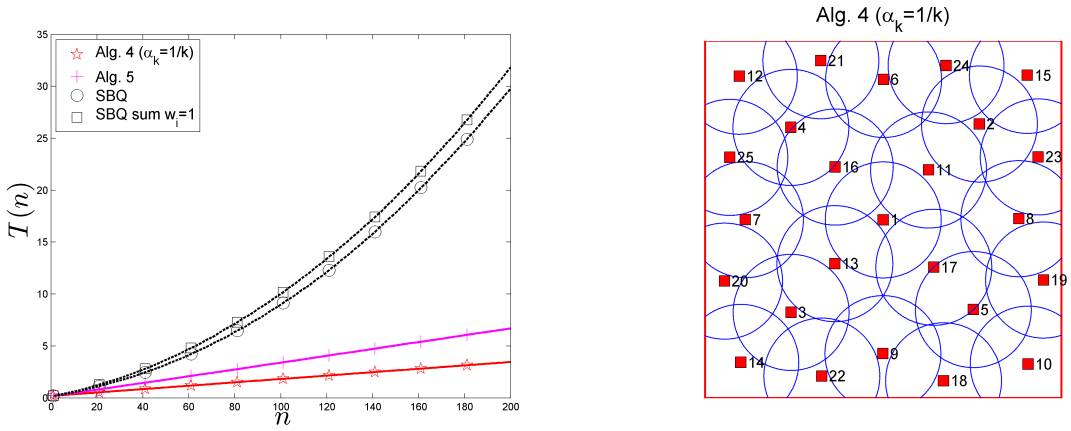

Figure 3: Left: computational time $T(n)$ (in s) of ξ_n for Algorithm 4 with $\alpha_k = 1/k$, Algorithm 5 and for the two versions (9) and (13) of SBQ, $n = 1, \ldots, 200$. Right: support $\mathbf{X}_n$ (ordered) of ξ_{25} generated with Algorithm 4 with $\alpha_k = 1/k$; the radius of the circles equals the covering radius of $\mathbf{X}_n$ (the smallest value that permits to cover $\mathscr{X}$).

7.2 Example 2: Gaussian mixture

Here $\mu = \sum_{j=1}^m \beta_j \, \mu_{\mathscr{N}}(\mathbf{a}_j, \sigma_j)$ with $\beta_j > 0$, $\sum_{j=1}^m \beta_j = 1$, where $\mu_{\mathscr{N}}(\mathbf{a}_j, \sigma_j)$ corresponds to the normal distribution with mean $\mathbf{a}_j$ and variance $\sigma_j^2 \, \mathbf{I}_d$, with $\mathbf{I}_d$ the identity matrix. Again, for illustration purpose, we take $d = 2$. The difficulty increases with the number m of components, the problem is also more difficult when the weights and/or variances of the components differ. We take $m = 3$, $\mathbf{a}_1 = (-1, 1)^\top$, $\mathbf{a}_2 = (1, -1)^\top$, $\mathbf{a}_3 = (1, 1)^\top$, $\sigma_j = 1/2$ for all j, and $\beta_1 = \beta_2 = 2/7$, $\beta_3 = 3/7$ (this is a slight variation of the example in Figure 1 of (Teymur et al., 2021) where the three components have equal weights). Figure 4 presents a 3-d plot of the probability density function φ_μ (left) and its contour lines (right) together with the candidate set $\mathscr{X}_C$ formed by $2^{14} = 16\,384$ independent samples, among which we shall select a subset of n representative points.

We use the Gaussian (or Radial Basis Function) kernel

$$K_\theta(\mathbf{x}, \mathbf{x}') = \exp -(\theta \, \|\mathbf{x} - \mathbf{x}'\|^2) \, , \tag{28}$$

$\mathsf{CR}(\mathbf{X}_n)$ indicates that for each point in $\mathscr{X}$ there is a design point at proximity, hence the frequent use of $\mathsf{CR}(\mathbf{X}_n)$ as a space-filling characteristic to be minimised.

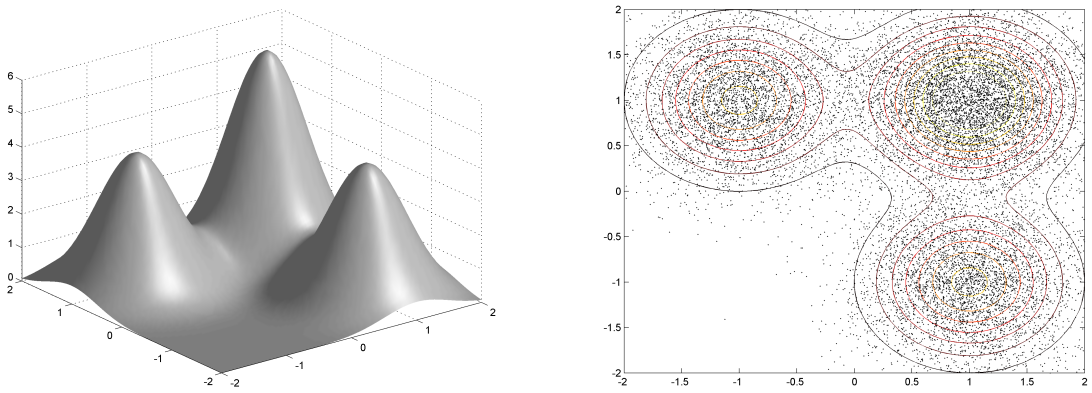

Figure 4: Left: 3d-plot of the p.d.f. φ_μ; right: contour lines of φ_μ and candidate set $\mathscr{X}_C$ (dots).

for which direct calculation gives[10]

$$\begin{aligned} \mathscr{E}_{K_\theta}(\mu) &= \sum_{j,\ell=1}^{d} \frac{\beta_j \beta_\ell}{(1+2\theta\sigma_j^2+2\theta\sigma_\ell^2)^{d/2}} \exp\left(-\frac{\theta\,\|\mathbf{a}_j-\mathbf{a}_\ell\|^2}{1+2\theta\sigma_j^2+2\theta\sigma_\ell^2}\right) \\ P_{K,\mu}(\mathbf{x}) &= \sum_{j=1}^{d} \frac{\beta_j}{(1+2\theta\sigma_j^2)^{d/2}} \exp\left(-\frac{\theta\,\|\mathbf{x}-\mathbf{a}_j\|^2}{1+2\theta\sigma_j^2}\right), \quad \mathbf{x}\in\mathbb{R}^d\,. \end{aligned} \tag{29}$$

It is important to choose a suitable order of magnitude for θ, even if a precise tuning is not essential. This issue is frequently mentioned in the literature, see for example Huszár and Duvenaud (2012), but is often overlooked. As for the construction of space-filling designs where the target measure μ is uniform on $\mathscr{X}$, see Pronzato and Zhigljavsky (2020), we recommend to let θ depend on the number of points to be generated. If the target size is $n_{\max}$ points, each point $\mathbf{x}_i$ will "represent" a fraction $1/n_{\max}$ of the C candidate points, and having a correlation $K_\theta(\mathbf{x}_i,\mathbf{x}) > 1/2$ with $C/n_{\max}$ points seems reasonable. We thus choose θ such that $K_\theta(\mathbf{x}_i,\mathbf{x}_j) < 1/2$ for $(100/n_{\max})\%$ of the pairs $(\mathbf{x}^{(j)},\mathbf{x}^{(k)})$ in a random sample of 1 000 points of $\mathscr{X}_C$ (that is, with obvious notation, $\theta = -\log(0.5)/Q_{1/n_{\max}}(\|\mathbf{x}^{(j)}-\mathbf{x}^{(k)}\|^2)$ for the example considered).

Figure 5 shows the first $n_{\max}$ points selected by Algorithms 1 and 4, both with $\alpha_k = 1/k$ for all k: $n_{\max} = 25$ ($\theta \simeq 5.7$) on the first row, $n_{\max} = 200$ ($\theta \simeq 46.4$) on the second. The points location looks roughly the same for both algorithms when $n_{\max} = 25$, the ordering being however different starting at $n = 12$; the designs look also similar when $n_{\max} = 200$ and it is difficult to separate them.

Figure 6 shows the evolution of $\mathsf{MMD}_K(\mu,\xi_n)$ and its upper bound (22) for ξ_n generated with Algorithms 1, 2, 3-(*ii*), 3-(*iii*), 4 and 5, and for empirical measures of random designs (empirical mean $\pm$ 2 standard deviations for 100 repetitions), when $n_{\max} = 200$ (the designs constructed algorithmically are not shown, but they all look very similar to those on the second row of

[10]The analytic expression of $P_{K,\mu}(\mathbf{x})$ is also available for K the product of uni-dimensional Matérn 3/2 kernels as in Section 7.1, though the expression is more complicated than (29) and involves the error function $\mathrm{erf}(t) = (2/\sqrt{\pi})\int_0^t \exp(-x^2)\mathrm{d}x$; the experimental results obtained are similar to those presented here for the Gaussian kernel.

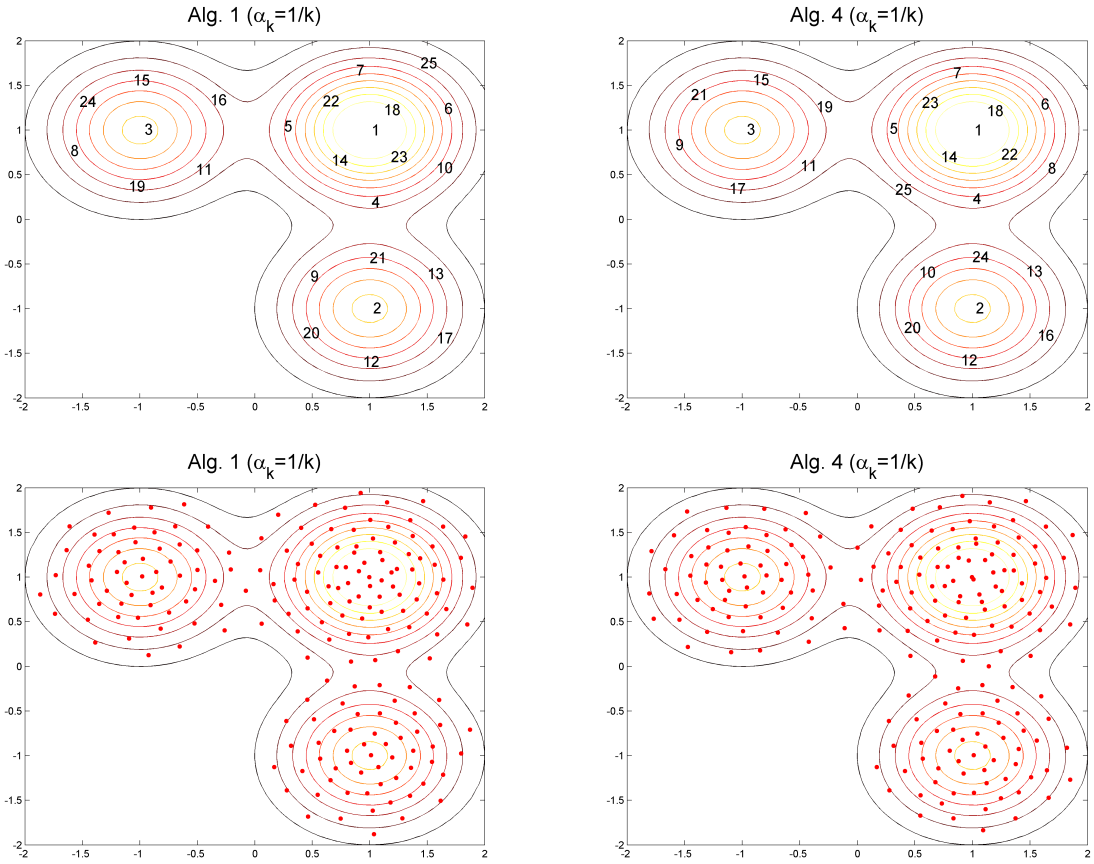

Figure 5: Designs $\mathbf{X}_n$ obtained with Algorithms 1 and 4 (both with $\alpha_k = 1/k$); $n_{\max} = 25$ ($\theta \simeq 5.7$) on the first row, $n_{\max} = 200$ ($\theta \simeq 46.39$) on the second row.

Figure 5). Algorithms 1 and 4 (both with $\alpha_k = 1/k$ for all k) and 2 and 5 perform similarly; Algorithm 3-(ii) is only marginally superior; the MMD is significantly smaller for Algorithm 3-(iii) which does not set constraints on ξ_n. The weights that ξ_n allocates to its support points are positive for Algorithm 3-(ii) and (iii), with $\sum_{i=1}^{200}\{\mathbf{w}_{200}\}_i \simeq 0.834$ for Algorithm 3-(iii). The two versions (9) and (13) of SBQ perform similarly to Algorithm 3-(iii) and (ii), respectively, and their weights $\{\mathbf{w}_{200}\}_i$ are positive too.

Finally, we also evaluate the approximation error by the MMD for the distance kernel $K_D(\mathbf{x}, \mathbf{x}') = -\|\mathbf{x} - \mathbf{x}'\|$ of Székely and Rizzo (2013). Since $P_{K_D,\mu}(\cdot)$ and the energy distance $\mathscr{E}_{K_D}(\mu)$ are not known explicitly, we compute $\mathsf{MMD}_{K_D}(\mu_C, \xi_n)$, with μ_C the empirical measure for the candidate set $\mathscr{X}_C$. Figure 7 shows $\mathsf{MMD}_{K_D}(\mu_C, \xi_n)$ for ξ_n generated with Algorithms 1, 3-(ii) and 4, using the Gaussian kernel (28)—ξ_n generated with Algorithm 3-(iii) cannot be tested since its weights do not sum to one[11]. The three algorithms appear to perform similarly and tend to provide better approximations of μ than random sampling in terms of $\mathsf{MMD}_{K_D}(\mu_C, \cdot)$ (the empirical mean $\pm$ two standard deviations for 100 random designs is presented). Due to

[11] K_D is Conditionally Integrally Strictly Positive Definite and defines a metric between probability measures, but $\mathscr{E}_{K_D}(\mu_C - \xi)$ can be negative for $\xi \notin \mathscr{M}_{[1]}(\mathscr{X})$.

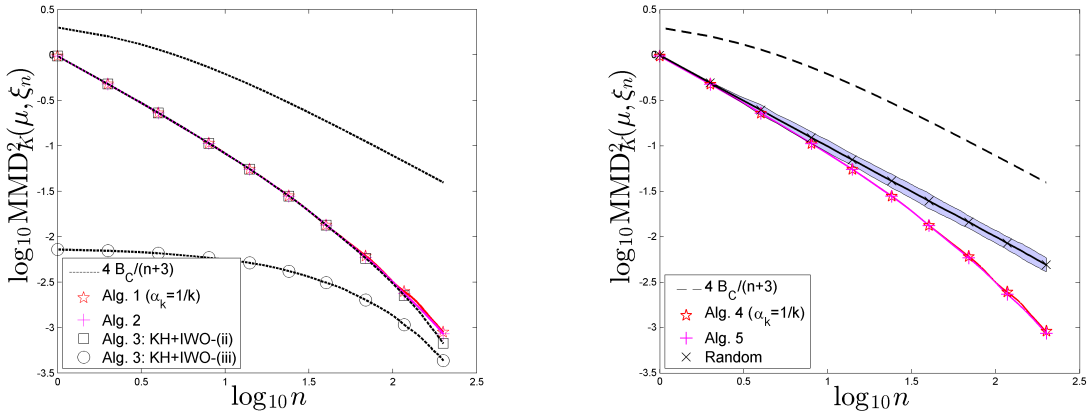

Figure 6: Upper bound (22) and $\mathsf{MMD}_K(\mu, \xi_n)$ for ξ_n generated with (left) Algorithms 1, 2, 3-(ii), 3-(iii); (right) Algorithms 4 and 5, and for the empirical measure of random designs (mean value $\pm$ 2σ over 100 repetitions).

their much smaller computational costs, Algorithms 1 and 4 are preferable to Algorithm 3-(ii) (and version (13) of SBQ) in this example. We also tried to use a Stein kernel, based on the inverse multiquadric kernel $K_{s,\theta}(\mathbf{x}, \mathbf{x}') = 1/(1 + \theta \|\mathbf{x} - \mathbf{x}'\|^2)^s$, $\theta > 0$, $s \in (0, 1)$, instead of (28) to generate ξ_n, but the values obtained for $\mathsf{MMD}_{K_D}(\mu_C, \xi_n)$ were significantly larger than with (28)[12].

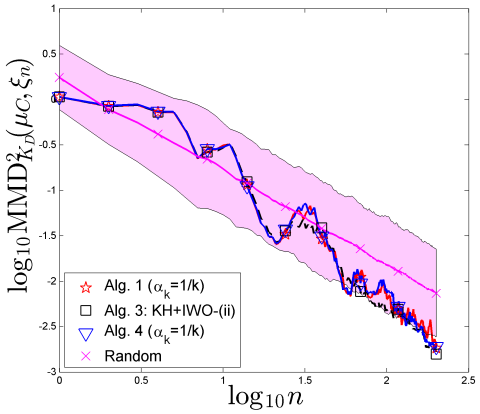

Figure 7: $\mathsf{MMD}_{K_D}(\mu_C, \xi_n)$ for the empirical measure of random designs (mean value $\pm$ 2σ over 100 repetitions) and for ξ_n generated with Algorithms 1, 3-(ii), and 4, all using the Gaussian kernel (28), with K_D the distance kernel of Székely and Rizzo (2013).

[12]We used $s = 1/2$ and θ given by the median heuristic of Garreau et al. (2017); see also Teymur et al. (2021).

8 Conclusions

Bounds on the finite-sample-size approximation error of iterative methods for the minimisation of an MMD discrepancy have been derived and illustrated by numerical experiments. These experiments indicate that the bounds give a fair picture of the decrease rate of the true MMD for some of the methods considered (Algorithms 2 and 5), but are pessimistic for most of them. This is particularly true for SBQ with unconstrained weights (9), for which the link with kernel herding used for the derivation of the error bound gives a plausible explanation for its marked pessimism; see Remark 5. These numerical results also indicate that the performances of kernel herding and greedy MMD minimisation do not improve by considering other step-size sequences than $1/k$ (which generate empirical measures), and that a variant of kernel herding with optimised weights, Algorithm 3-(*iii*), yields performance similar to standard SBQ for a slightly lower computational cost. Therefore, on the whole, Algorithm 3-(*iii*) appears to be the best option when the budget n is very limited (its complexity is quadratic in n), and standard KH or MMD, with uniform weights, seem generally preferable to more sophisticated methods for large n (their complexity is linear in n).

We have restricted our attention to finite candidate sets. This situation is at the same time easier and computationally more efficient in terms of practical implementation, and simpler in terms of analysis since only finite-dimensional linear algebra is used, but the extension to the Hilbert-space situation remains possible.

Finally, we have only considered the case where one adds one-point-at-a-time to the construction. Less myopic methods that select several (say $m > 1$) points at each iterations could also be considered. The extension of our results to this context, and the development of computationally efficient methods that avoid the combinatorial explosion due to considering all possible $\binom{C}{m}$ subset selections, deserve further studies. One can refer to Teymur et al. (2021) for an exciting contribution in this direction.

Appendix A: alternative expressions for $\mathsf{MMD}_K^2(\mu, \xi)$

The quadratic form (4) of $\mathsf{MMD}_K^2(\mu, \xi_n)$ and the fact that $\widetilde{\xi}_n$ has unconstrained optimal weights implies that, for any $\xi_n \in \mathscr{M}(\mathbf{X}_n)$,

$$\mathsf{MMD}_K^2(\mu, \xi_n) = (\mathbf{w}_n - \widetilde{\mathbf{w}}_n)^\top \mathbf{K}_n (\mathbf{w}_n - \widetilde{\mathbf{w}}_n) + \mathsf{MMD}_K^2(\mu, \widetilde{\xi}_n)\,.$$

Therefore, any measure ξ in $\mathscr{M}(\mathscr{X}_C)$ with associated weights ω satisfies

$$\mathsf{MMD}_K^2(\mu, \xi) = \widetilde{g}_C(\omega) + \mathsf{MMD}_K^2(\mu, \widetilde{\xi}^C)\,, \tag{30}$$

where, for any $\omega \in \mathbb{R}^C$, we denote

$$\widetilde{g}_C(\omega) = \|\omega - \widetilde{\omega}^C\|_{\mathbf{K}_C}^2 = (\omega - \widetilde{\omega}^C)^\top \mathbf{K}_C (\omega - \widetilde{\omega}^C) \quad (= \mathsf{MMD}_K^2(\xi, \widetilde{\xi}^C))\,. \tag{31}$$

When $\xi_n \in \mathscr{M}_{[1]}(\mathbf{X}_n)$ (i.e., $\mathbf{w}_n^\top \mathbf{1}_n = 1$), as $(\mathbf{w}_n - \widehat{\mathbf{w}}_n)^\top \mathbf{1}_n = 0$ and $\mathbf{K}_n(\widehat{\mathbf{w}}_n - \widetilde{\mathbf{w}}_n) \propto \mathbf{1}_n$, see (5) and (6),

$$\begin{aligned} \mathsf{MMD}_K^2(\mu, \xi_n) &= (\mathbf{w}_n - \widehat{\mathbf{w}}_n + \widehat{\mathbf{w}}_n - \widetilde{\mathbf{w}}_n)^\top \mathbf{K}_n (\mathbf{w}_n - \widehat{\mathbf{w}}_n + \widehat{\mathbf{w}}_n - \widetilde{\mathbf{w}}_n) + \mathsf{MMD}_K^2(\mu, \widetilde{\xi}_n)\,, \\ &= (\mathbf{w}_n - \widehat{\mathbf{w}}_n)^\top \mathbf{K}_n (\mathbf{w}_n - \widehat{\mathbf{w}}_n) + \mathsf{MMD}_K^2(\mu, \widehat{\xi}_n)\,. \end{aligned}$$

Therefore, any measure ξ in $\mathscr{M}_{[1]}(\mathscr{X}_C)$ with associated weights ω satisfies

$$\mathsf{MMD}_K^2(\mu, \xi) = \widehat{g}_C(\omega) + \mathsf{MMD}_K^2(\mu, \widehat{\xi}^C)\,, \tag{32}$$

where

$$\widehat{g}_C(\omega) = \|\omega - \widehat{\omega}^C\|^2_{\mathbf{K}_C} = (\omega - \widehat{\omega}^C)^\top \mathbf{K}_C (\omega - \widehat{\omega}^C) \quad (= \mathsf{MMD}^2_K(\xi, \widehat{\xi}^C)) \,. \tag{33}$$

For any measure $\xi \in \mathscr{M}(\mathscr{X}_C)$ with associated weights $\omega \in \mathscr{P}_C$, we define

$$\Delta_C(\xi) = \mathsf{MMD}^2_K(\mu, \xi) - M_C^2 \,, \tag{34}$$

so that (30) implies $\Delta_C(\xi) = \widetilde{g}_C(\omega) - \widetilde{g}_C(\omega_*^C)$ and, when $\xi \in \mathscr{M}_{[1]}(\mathscr{X}_C)$, (32) implies $\Delta_C(\xi) = \widehat{g}_C(\omega) - \widehat{g}_C(\omega_*^C)$.

Appendix B: proofs

Our derivations of bounds on $\Delta_C(\xi_k)$ given by (34) (i.e., on $\mathsf{MMD}^2_K(\mu, \xi_k)$) rely on the convexity of $\widehat{g}_C(\cdot)$ and $\widetilde{g}_C(\cdot)$ and on the following lemma.

Lemma 2. *Let $(t_k)_k$ and $(\alpha_k)_k$ be two real positive sequences and A be a strictly positive real.*
(i) If t_k satisfies

$$t_1 \le A \quad \textit{and} \quad t_{k+1} \le (1 - \alpha_{k+1})\, t_k + A\, \alpha_{k+1}^2 \,, \ k \ge 1 \,, \tag{35}$$

with $\alpha_k = 1/k$ for all k, then $t_k \le A\,(2 + \log k)/(k+1)$ for all $k \ge 1$.
(ii) If t_k satisfies (35) with $\alpha_k = 2/(k+1)$ for all k, then $t_k \le 4\,A/(k+3)$ for all $k \ge 1$.
(iii) If t_k satisfies

$$t_1 \le A \quad \textit{and} \quad t_{k+1} \le (1 - 2\,\alpha_{k+1})\, t_k + A\, \alpha_{k+1}^2 \,, \ k \ge 1 \,, \tag{36}$$

with $\alpha_k = 1/k$ for all k, then $t_k \le A/k$ for all $k \ge 1$.
(iv) If t_k satisfies

$$t_1 \le A \quad \textit{and} \quad t_{k+1} \le t_k - \frac{t_k^2}{A} \,, \ k \ge 1 \,, \tag{37}$$

then, $t_k \le A/(k + p_2)$ for all $k \ge 2$, with $p_2 = A/t_2 - 2 \ge 2$; moreover, when $t_1 \le A/2$, $t_k \le A/(k + p_1)$ for all $k \ge 1$, with $p_1 = A/t_1 - 1 \ge 1$.

Proof.

(*i*) Suppose that t_k satisfies (35) with $\alpha_k = 1/k$ for all k. We show that $t_k \le A\,(2 + \log k)/(k+1)$ by induction on k. The inequality is satisfied for $k = 1$, assume that it is satisfied for $k \ge 1$. We get $A\,[2 + \log(k+1)]/(k+2) - t_{k+1} \ge A\,a(k)/[(k+2)(k+1)^2]$, with $a(k) = (k+1)^2\, \log(1 + 1/k) + \log k - k \ge 0$, implying that the inequality is satisfied for all k.

(*ii*) Suppose now that t_k satisfies (35) with $\alpha_{k+1} = b/(k + 1 + q)$ for all k, for some $0 < b < q + 2$. We prove that $t_k \le A\,a/(k+p)$ for some $a, p > 0$ by induction on k. Not all values of a, b, p, q are legitimate, and a natural objective is to have a and p respectively as small and large as possible. We show that the best choice is that indicated in Lemma 2. For $k = 1$, since $t_1 \le A$, to ensure that $t_1 \le A\,a/(p+1)$ we need to have $p \le a - 1$. Assume that $t_k \le A\,a/(k+p)$, and denote $\delta_k = A\,a/(k+1+p) - t_{k+1}$. It satisfies

$$\delta_k \quad \ge \quad A\,\frac{a}{k+1+p} - \left[(1 - \alpha_{k+1})\, t_k + B_C\, \alpha_{k+1}^2\right] \ge A\,\frac{a(k)}{(k+p)(k+1+p)(k+1+q)^2} \,,$$

where $a(k)$ is a second-degree polynomial in k, with leading term $(ab - a - b^2)\, k^2$. We thus need to choose a pair (a, b) of positive numbers such that $ab - a - b^2 \ge 0$; the pair with the smallest value of a is $(4, 2)$. For this choice of a, b, we get $a(k) = 4\,[k + 1 + p - (p - q)^2]$, which increases with k. We only need to guarantee that $a(1) \ge 0$, which corresponds to $q + 1/2 - (1/2)\,\sqrt{4q + 9} \le p \le q + 1/2 + (1/2)\,\sqrt{4q + 9}$.

Since $a = 4$, the largest p allowed is $p = 3$, which is admissible for $q = 1$. In that case, $a(k) = 4k$ and $\delta_k \geq 0$, showing that $t_k \leq 4\,A/(k+3)$ for all k when (35) is satisfied with $\alpha_k = 2/(k+1)$ for all k.

(*iii*) Suppose that t_k satisfies (36) with $\alpha_k = 1/k$ for all k. We have $t_1 \leq A$ by hypothesis; the induction hypothesis $t_k \leq A/k$ and (36) give $A/(k+1) - t_{k+1} \geq A/[k(k+1)^2] > 0$, and thus imply that $t_{k+1} \leq A/(k+1)$.

(*iv*) The function $t \to f(t) = t - t^2/A$ is increasing on $[0, A/2)$ with a maximum on $[0, A]$ equal to $A/4$ attained for $t = A/2$. We have $t_2 \leq f(A/2) = A/4$, and thus $t_k \leq A/4$ for all $k \geq 2$. Take $p = A/t_2 - 2$, so that $p \geq 2$ and $t_2 = A/(p+2) \leq A/4$. Suppose that $t_k \leq A/(p+k)$; we have $t_{k+1} \leq A\,[1/(k+p) - 1/(k+p)^2] = A\,(k+p-1)/(k+p)^2 = A/(k+p+1) - A/[(k+p)^2(k+p+1)] < A/(k+p+1)$, showing that $t_k \leq A/(p+k)$ for all $k \geq 2$.

When $t_1 \leq A/2$, we take $p = A/t_1 - 1$, which gives $p \geq 1$, $t_1 = A/(p+1) \leq A/2$ and $t_k < A/2$ for all $k > 1$. Assuming that $t_k \leq A/(p+k)$, we get $t_{k+1} < A/(k+p+1)$, showing that $t_k \leq A/(p+k)$ for all $k \geq 1$. ∎

Proof of Theorem 1. The proof is based on (Clarkson, 2010). For any $\mathbf{x}^{(j)} \in \mathscr{X}_C$, the definition (34) of $\Delta_C(\xi)$ gives

$$\begin{aligned}\Delta_C[\xi_k^+(\mathbf{x}^{(j)}, \alpha_{k+1})] &= \widehat{g}_C[\omega_k + \alpha_{k+1}(\mathbf{e}_j - \omega_k)] - \widehat{g}_C(\omega_*^C) \\ &= \widehat{g}_C(\omega_k) - \widehat{g}_C(\omega_*^C) + 2\,\alpha_{k+1}\,(\mathbf{e}_j - \omega_k)^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C) + \alpha_{k+1}^2 \|\mathbf{e}_j - \omega_k\|_{\mathbf{K}_C}^2\,, \quad (38)\end{aligned}$$

where $\alpha_{k+1} = 1/(k+1)$. The definition of $\widehat{\omega}^C$ implies that $\mathbf{u}^\top \mathbf{K}_c \widehat{\omega}^C = \mathbf{u}^\top \mathbf{p}_C(\mu)$ for any $\mathbf{u} \in \mathbb{R}^C$ orthogonal to $\mathbf{1}_C$, see (5). Therefore, $(\mathbf{e}_j - \omega_k)^\top \mathbf{K}_C \widehat{\omega}^C = P_{K,\mu}(\mathbf{x}^{(j)}) - \sum_{i=1}^k \{\mathbf{w}_k\}_i P_{K,\mu}(\mathbf{x}_i)$, and $\mathbf{x}_{k+1} = \mathbf{x}^{(j_{k+1})}$ with

$$j_{k+1} \in \operatorname{Arg\,min}_{j \in \mathbb{I}_C} P_{K,\xi_k}(\mathbf{x}^{(j)}) - P_{K,\mu}(\mathbf{x}^{(j)}) = \operatorname{Arg\,min}_{j \in \mathbb{I}_C} (\mathbf{e}_j - \omega_k)^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C)\,,$$

in agreement with (16). The convexity of $\widehat{g}_C(\cdot)$ implies that

$$\widehat{g}_C(\omega_*^C) \geq \widehat{g}_C(\omega_k) + 2\,(\omega_*^C - \omega_k)^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C) \geq \widehat{g}_C(\omega_k) + 2 \min_{j \in \mathbb{I}_C} (\mathbf{e}_j - \omega_k)^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C)\,, \quad (39)$$

where the second inequality follows from $\omega_*^C \in \mathscr{P}_C$, implying that

$$\Delta_C(\xi_{k+1}) = \Delta_C[\xi_k^+(\mathbf{x}_{k+1}, \alpha_{k+1})] \quad \leq \quad (1 - \alpha_{k+1})\,[\widehat{g}_C(\omega_k) - \widehat{g}_C(\omega_*^C)] + \alpha_{k+1}^2 \|\mathbf{e}_{j_{k+1}} - \omega_k\|_{\mathbf{K}_C}^2\,.$$

The last term can be bounded as follows: as the weights ω_k belong to $\mathscr{P}_C$ for all k, for all $j \in \mathbb{I}_C$, we have

$$\begin{aligned}\|\mathbf{e}_j - \omega_k\|_{\mathbf{K}_C}^2 \leq \max_{\omega, \omega' \in \mathscr{P}_C} (\omega - \omega')^\top \mathbf{K}_C(\omega - \omega') &= \max_{i,j \in \mathbb{I}_C} (\mathbf{e}_i - \mathbf{e}_j)^\top \mathbf{K}_C(\mathbf{e}_i - \mathbf{e}_j) \\ &= \max_{i,j \in \mathbb{I}_C} K(\mathbf{x}^{(i)}, \mathbf{x}^{(i)}) + K(\mathbf{x}^{(j)}, \mathbf{x}^{(j)}) - 2\,K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \leq B_C\,, \quad (40)\end{aligned}$$

with $B_C = 4\,\overline{K}_C$ ($B_C = 2\,\overline{K}_C$ when $K \geq 0$). It implies that

$$\Delta_C(\xi_{k+1}) \leq (1 - \alpha_{k+1})\,\Delta_C(\xi_k) + B_C\,\alpha_{k+1}^2\,. \quad (41)$$

Since

$$\Delta_C(\xi_1) \leq \mathsf{MMD}_K^2(\mu, \xi_1) = K(\mathbf{x}_1, \mathbf{x}_1) - 2\,P_{K,\mu}(\mathbf{x}_1) + \mathscr{E}_K(\mu) \leq B_C\,, \quad (42)$$

Lemma 2-(i) gives (20) when $\alpha_k = 1/k$ for all k.

When $\widehat{\xi}^C = \xi_*^C$, $\widehat{\omega}^C \in \mathscr{P}_C$, and $\Delta_C(\xi_k) = \widehat{g}_C(\omega_k)$. Therefore,

$$\min_{j \in \mathbb{I}_C} (\mathbf{e}_j - \omega_k)^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C) \leq (\widehat{\omega}^C - \omega_k)^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C) = -\widehat{g}_C(\omega_k)\,, \quad (43)$$

and we get $\Delta_C(\xi_{k+1}) \leq (1 - 2\,\alpha_{k+1})\,\Delta_C(\xi_k) + B_C\,\alpha_{k+1}^2$ instead of (41). Lemma 2-(*iii*) gives (21). ■

Proof of Theorem 2. $\Delta_C(\xi_k)$ satisfies (41) with $\alpha_{k+1} = 2/(k+2)$. Lemma 2-(*ii*) gives (22). ■

Proof of Theorem 3. Consider again the proof of Theorem 1. We have

$$\Delta_C(\xi_{k+1}) = \min_{\alpha\in[0,1]} \Delta_C[\xi_k^+(\mathbf{x}^{(j_{k+1})}, \alpha)] \leq \Delta_C[\xi_k^+(\mathbf{x}^{(j_{k+1})}, \alpha_{k+1})] \leq (1 - \alpha_{k+1})\,\Delta_C(\xi_k) + B_C\,\alpha_{k+1}^2\,, \quad (44)$$

for any arbitrary choice of $\alpha_{k+1} \in [0,1]$, and $\alpha_{k+1} = 2/(k+2)$ for all k has been shown to imply (22) in Theorem 2. When $\widehat{\xi}^C = \xi_*^C$, we get $\Delta_C(\xi_{k+1}) \leq (1 - 2\,\alpha_{k+1})\,\Delta_C(\xi_k) + B_C\,\alpha_{k+1}^2$, see the proof of Theorem 1. When we use $\alpha_k = 1/k$ for all k, Lemma 2-(*iii*) implies (21).

The value of α minimising $\Delta_C[\xi_k^+(\mathbf{x}^{(j_{k+1})}, \alpha)]$ is

$$\widehat{\alpha}_{k+1} = \frac{(\omega_k - \mathbf{e}_{j_{k+1}})^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C)}{\|\mathbf{e}_{j_{k+1}} - \omega_k\|_{\mathbf{K}_C}^2}\,, \quad (45)$$

so that (39) implies $\widehat{\alpha}_{k+1} > 0$ when $\mathsf{MMD}_K(\mu, \xi_k) > \mathsf{MMD}_K(\mu, \xi_*^C)$. The algorithm can thus be stopped if $\widehat{\alpha}_{k+1} = 0$. Since $(\omega_k - \mathbf{e}_{j_{k+1}})^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C) = \|\mathbf{e}_{j_{k+1}} - \omega_k\|_{\mathbf{K}_C}^2 + (\mathbf{e}_{j_{k+1}} - \widehat{\omega}^C)^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C) - \|\mathbf{e}_{j_{k+1}} - \widehat{\omega}^C\|_{\mathbf{K}_C}^2$, we have $\widehat{\alpha}_{k+1} \leq 1 + (\mathbf{e}_{j_{k+1}} - \widehat{\omega}^C)^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C)/\|\mathbf{e}_{j_{k+1}} - \omega_k\|_{\mathbf{K}_C}^2$. When $\widehat{\omega}^C \in \mathscr{P}_C$ (that is, when $\widehat{\omega}^C = \omega_*^C$), the definition of j_{k+1} implies that $\widehat{\alpha}_{k+1} \leq 1$ (since $\sum_{j=1}^C \{\widehat{\omega}^C\}_j (\mathbf{e}_j - \widehat{\omega}^C)^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C) = 0$ with all $\{\widehat{\omega}^C\}_j \geq 0$). However, nothing guarantees that $\widehat{\alpha}_{k+1} \leq 1$ in general. Direct calculation gives

$$(\omega_k - \mathbf{e}_j)^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C) = \sum_{i,\ell=1}^k \{\mathbf{w}_k\}_i \{\mathbf{w}_k\}_\ell K(\mathbf{x}_i, \mathbf{x}_\ell) - \sum_{i=1}^k \{\mathbf{w}_k\}_i K(\mathbf{x}_i, \mathbf{x}^{(j)}) + P_{K,\mu}(\mathbf{x}^{(j)}) - \sum_{i=1}^k \{\mathbf{w}_k\}_i P_{K,\mu}(\mathbf{x}_i)\,, \quad (46)$$

$$\|\mathbf{e}_j - \omega_k\|_{\mathbf{K}_C}^2 = \sum_{i,\ell=1}^k \{\mathbf{w}_k\}_i \{\mathbf{w}_k\}_\ell K(\mathbf{x}_i, \mathbf{x}_\ell) - 2\sum_{i=1}^k \{\mathbf{w}_k\}_i K(\mathbf{x}_i, \mathbf{x}^{(j)}) + K(\mathbf{x}^{(j)}, \mathbf{x}^{(j)})\,, \quad (47)$$

which together with (45) gives (23). The recursive updating of $Q_k = \sum_{i,\ell=1}^k \{\mathbf{w}_k\}_i \{\mathbf{w}_k\}_\ell K(\mathbf{x}_i, \mathbf{x}_\ell)$, $R_k = \sum_{i=1}^k \{\mathbf{w}_k\}_i P_{K,\mu}(\mathbf{x}_i)$ and $S_k(\mathbf{x}) = P_{K,\xi_k}(\mathbf{x})$ gives Algorithm 2. ■

Proof of Theorem 4.

(*i*) When $\nu_k = \xi_k^*$ is substituted for ξ_k, we have $\mathbf{x}_{k+1} = \mathbf{x}^{(j_{k+1})}$ with $j_{k+1} \in \operatorname{Arg\,min}_{j\in\mathbb{I}_C} (\mathbf{e}_j - \omega_k^*)^\top \mathbf{K}_C(\omega_k^* - \widehat{\omega}^C)$ and, by construction, $\Delta_C(\xi_{k+1}) \leq \Delta_C[\nu_k^+(\mathbf{x}_{k+1}, \alpha)]$ for any $\alpha \in [0,1]$, where $\Delta_C(\xi)$ is given by (34) and $\nu_k^+(\mathbf{x}, \alpha) = (1-\alpha)\,\xi_k^* + \alpha\,\delta_{\mathbf{x}}$. Consider (38) in the proof of Theorem 1: we have

$$\Delta_C[\nu_k^+(\mathbf{x}_{k+1}, \alpha)] = \widehat{g}_C(\omega_k^*) - \widehat{g}_C(\omega_*^C) + 2\,\alpha\,(\mathbf{e}_{j_{k+1}} - \omega_k^*)^\top \mathbf{K}_C(\omega_k^* - \widehat{\omega}^C) + \alpha^2 \|\mathbf{e}_{j_{k+1}} - \omega_k^*\|_{\mathbf{K}_C}^2\,,$$

and the convexity of $\widehat{g}_C(\cdot)$ implies

$$\widehat{g}_C(\omega_*^C) \geq \widehat{g}_C(\omega_k^*) + 2\,(\omega_*^C - \omega_k^*)^\top \mathbf{K}_C(\omega_k^* - \widehat{\omega}^C) \geq \widehat{g}_C(\omega_k^*) + 2\,(\mathbf{e}_{j_{k+1}} - \omega_k^*)^\top \mathbf{K}_C(\omega_k^* - \widehat{\omega}^C)\,,$$

where the second inequality follows from $\omega_*^C \in \mathscr{P}_C$ and the choice of j_{k+1}. Using $\|\mathbf{e}_{j_{k+1}} - \omega_k^*\|_{\mathbf{K}_C}^2 \leq B_C$, see the proof of Theorem 1, we thus obtain

$$\Delta_C(\xi_{k+1}) \leq \Delta_C[\nu_k^+(\mathbf{x}_{k+1}, \alpha_{k+1})] \leq (1 - \alpha_{k+1})\,\Delta_C(\nu_k) + B_C\,\alpha_{k+1}^2\,, \quad k \geq 1\,, \quad (48)$$

for any predefined α_{k+1}. When $\alpha_{k+1} = 2/(k+2)$, the induction used in the proof of Theorem 2 gives (22). When $\widehat{\xi}^C = \xi_*^C$, the right-hand side of (48) becomes $(1 - 2\,\alpha_{k+1})\,\Delta_C(\nu_k) + B_C\,\alpha_{k+1}^2$, see the proof of Theorem 1, and if we take $\alpha_k = 1/k$ for all k, Lemma 2-(*iii*) implies (21).

(*ii*) Suppose now that $\nu_k = \widehat{\xi}_k$ is substituted for ξ_k at iteration k. Equation (46) gives

$$P_{K,\widehat{\xi}_k}(\mathbf{x}_{k+1}) - P_{K,\mu}(\mathbf{x}_{k+1}) + \widehat{\mathbf{w}}_k^\top \mathbf{p}_k(\mu) - \mathscr{E}_K(\widehat{\xi}_k) = (\mathbf{e}_{j_{k+1}} - \widehat{\omega}_k)^\top \mathbf{K}_C(\widehat{\omega}_k - \widehat{\omega}^C)\,,$$

so that (12) gives

$$\Delta_C(\xi_{k+1}) = \Delta_C(\xi_k) - \frac{\left[(\mathbf{e}_{j_{k+1}} - \widehat{\omega}_k)^\top \mathbf{K}_C(\widehat{\omega}_k - \widehat{\omega}^C)\right]^2}{\min\limits_{\substack{\mathbf{w}\,\in\,\mathbb{R}^k \\ \mathbf{1}_k^\top \mathbf{w} = 1}} \|K(\mathbf{x}_{k+1},\cdot) - \mathbf{w}^\top \mathbf{k}_k(\cdot)\|^2_{\mathcal{H}_K}}\,.$$

As long as $\widehat{g}_C(\omega_*^C) \le \widehat{g}_C(\widehat{\omega}_k)$, that is, $\Delta_C(\widehat{\xi}_k) \ge 0$, (39) with $\widehat{\omega}_k$ substituted for ω_k gives

$$\begin{aligned}
\Delta_C(\xi_{k+1}) &\le \Delta_C(\xi_k) - \frac{\left[\widehat{g}_C(\widehat{\omega}_k) - \widehat{g}_C(\omega_*^C)\right]^2}{4\,\|K(\mathbf{x}_{k+1},\cdot) - (\mathbf{1}_k^\top/k)\mathbf{k}_k(\cdot)\|^2_{\mathcal{H}_K}} \\
&= \Delta_C(\xi_k) - \frac{\Delta_C^2(\xi_k)}{4\,\left[K(\mathbf{x}_{k+1},\mathbf{x}_{k+1}) + \mathbf{1}_k^\top \mathbf{K}_k \mathbf{1}_k/k^2 - 2\,\mathbf{1}_k^\top \mathbf{k}_k(\mathbf{x}_{k+1})/k\right]} \\
&\le \Delta_C(\xi_k) - \frac{\Delta_C^2(\xi_k)}{4\,B_C}\,, \qquad (49)
\end{aligned}$$

with $\Delta_C(\xi_1) \le B_C$, see (42). Lemma 2-(*iv*) with $A = 4\,B_C$ and $p_1 = 3$ gives (22). When $\widehat{\xi}^C = \xi_*^C$, (43) gives $\Delta_C(\xi_{k+1}) \le \Delta_C(\xi_k) - \Delta_C^2(\xi_k)/B_C$, and Lemma 2-(*iv*) with $A = B_C$ and $p_2 = 2$ gives (24).

(*iii*) Suppose finally that $\nu_k = \widetilde{\xi}_k$ is substituted for ξ_k at iteration k. Since $\widetilde{\xi}_k$ is not necessarily in $\mathscr{M}_{[1]}(\mathscr{X}_C)$, we shall use (31) instead of (33). The definition of $\widetilde{\omega}^C$ implies $\mathbf{K}_C\widetilde{\omega}^C = \mathbf{p}_C(\mu)$, and thus $\mathbf{x}_{k+1} = \mathbf{x}^{(j_{k+1})}$ with

$$j_{k+1} \in \operatorname{Arg\,min}_{j\,\in\,\mathbb{I}_C} P_{K,\nu_k}(\mathbf{x}^{(j)}) - P_{K,\mu}(\mathbf{x}^{(j)}) = \operatorname{Arg\,min}_{j\,\in\,\mathbb{I}_C} \mathbf{e}_j^\top \mathbf{K}_C(\widetilde{\omega}_k - \widetilde{\omega}^C)\,.$$

Let $\widetilde{\mathbf{g}}_k = 2\,\mathbf{K}_C(\widetilde{\omega}_k - \widetilde{\omega}^C)$ denote the gradient of $\widetilde{g}(\cdot)$ at $\widetilde{\omega}_k$. By construction, $\mathbf{e}_j^\top \widetilde{\mathbf{g}}_k = 0$ for all j with $\mathbf{x}^{(j)} \in \operatorname{supp}(\widehat{\xi}^k)$, so that $\widetilde{\omega}_k^\top \mathbf{K}_C(\widetilde{\omega}_k - \widetilde{\omega}^C) = 0$, and the convexity of $\widetilde{g}(\cdot)$ implies

$$\widetilde{g}_C(\omega_*^C) \ge \widetilde{g}_C(\widetilde{\omega}_k) + 2\,(\omega_*^C - \widetilde{\omega}_k)^\top \mathbf{K}_C(\widetilde{\omega}_k - \widetilde{\omega}^C) = \widetilde{g}_C(\widetilde{\omega}_k) + 2\,{\omega_*^C}^\top \mathbf{K}_C(\widetilde{\omega}_k - \widetilde{\omega}^C)\,. \qquad (50)$$

Since $\omega_*^C \in \mathscr{P}_C$, the definition of j_{k+1} implies

$$\widetilde{g}_C(\omega_*^C) \ge \widetilde{g}_C(\widetilde{\omega}_k) + 2\,\mathbf{e}_{j_{k+1}}^\top \mathbf{K}_C(\widetilde{\omega}_k - \widetilde{\omega}^C)\,. \qquad (51)$$

Therefore, as long as $\Delta_C(\widetilde{\xi}_k) = \widetilde{g}_C(\widetilde{\omega}_k) - \widetilde{g}_C(\omega_*^C) \ge 0$, (8) gives

$$\begin{aligned}
\Delta_C(\xi_{k+1}) &= \Delta_C(\xi_k) - \frac{\left[\mathbf{e}_{j_{k+1}}^\top \mathbf{K}_C(\widetilde{\omega}_k - \widetilde{\omega}^C)\right]^2}{\left[K(\mathbf{x}_{k+1},\mathbf{x}_{k+1}) - \mathbf{k}_k^\top(\mathbf{x}_{k+1})\mathbf{K}_k^{-1}\mathbf{k}_k(\mathbf{x}_{k+1})\right]} \\
&\le \Delta_C(\xi_k) - \frac{\Delta_C^2(\xi_k)}{4\,\left[K(\mathbf{x}_{k+1},\mathbf{x}_{k+1}) - \mathbf{k}_k^\top(\mathbf{x}_{k+1})\mathbf{K}_k^{-1}\mathbf{k}_k(\mathbf{x}_{k+1})\right]} \\
&\le \Delta_C(\xi_k) - \frac{\Delta_C^2(\xi_k)}{4\,\overline{K}_C} \le \Delta_C(\xi_k) - \frac{\Delta_C^2(\xi_k)}{4\,\overline{K}}\,. \qquad (52)
\end{aligned}$$

Since any signed measure supported on $\mathscr{X}_C$ can be used, we may start with $\xi_0 = 0$, with weights $\omega_0 = \mathbf{0}_C$, so that (8) and the inequality above apply from $k = 0$. We have $\Delta_C(\xi_0) \le \mathsf{MMD}_K^2(\mu,\xi_0) = \mathscr{E}_K(\mu) \le \overline{K}$; therefore, $\Delta_C(\xi_k) \le 4\,\overline{K}/(k+p_1)$, $k \ge 1$, for $p_1 = 4\,\overline{K}/\mathsf{MMD}_K^2(\mu,\xi_1) - 1 \le 4\,\overline{K}/\Delta_C(\xi_1) - 1$; see the proof

of Lemma 2-(*iv*). $\Delta_C(\xi_0) \leq \overline{K}$ implies $\Delta_C(\xi_1) \leq 3\,\overline{K}/4$, and we can also take $p_1 = 13/3$ in Lemma 2-(*iv*), which gives $\Delta_C(\xi_k) \leq 4\,\overline{K}/(k+13/3)$, $k \geq 1$. This completes the proof of (25). ∎

Stopping conditions for Algorithms 3-(ii) and (iii). Let $\widehat{\mathbf{g}}_k = 2\,\mathbf{K}_C(\widehat{\omega}_k - \widehat{\omega}^C)$ denote the gradient of $\widehat{g}(\cdot)$ at $\widehat{\omega}_k$. By construction, $(\mathbf{e}_j - \widehat{\omega}_k)^\top \widehat{\mathbf{g}}_k = 0$ for all j such that $\mathbf{x}^{(j)} \in \mathrm{supp}(\widehat{\xi}^k)$, with

$$(\mathbf{e}_j - \widehat{\omega}_k)^\top \widehat{\mathbf{g}}_k = 2\left\{ P_{K,\widehat{\xi}^k}(\mathbf{x}^{(j)}) - P_{K,\mu}(\mathbf{x}^{(j)}) + \sum_{i=1}^{k} \{\widehat{\mathbf{w}}_k\}_i \left[P_{K,\mu}(\mathbf{x}_i) - P_{K,\widehat{\xi}^k}(\mathbf{x}_i) \right] \right\}.$$

Therefore, $P_{K,\widehat{\xi}^k}(\mathbf{x}^{(j)}) - P_{K,\mu}(\mathbf{x}^{(j)})$ equals a constant c_k for all $\mathbf{x}^{(j)} \in \mathrm{supp}(\widehat{\xi}^k)$, and the existence of an $\mathbf{x} \in \mathscr{X}_C$ such that $P_{K,\widehat{\xi}^k}(\mathbf{x}) - P_{K,\mu}(\mathbf{x}) < c_k$ implies that Algorithm 3-(*ii*) can still progress. Conversely, if $P_{K,\widehat{\xi}^k}(\mathbf{x}) - P_{K,\mu}(\mathbf{x}) \geq c_k$ for all $\mathbf{x} \in \mathscr{X}_C$, the algorithm can be stopped. We can thus add the following line to Algorithm 3-(*ii*):

4'-(*ii*): if $S_{k-1}(\mathbf{x}_k) - P_{K,\mu}(\mathbf{x}_k) \geq S_{k-1}(\mathbf{x}_{k-1}) - P_{K,\mu}(\mathbf{x}_{k-1})$ **then return** $\mathbf{X}_{k-1}$, ξ_{k-1} and stop;

Similarly, $\mathbf{e}_j^\top \widetilde{\mathbf{g}}_k = 2\left[P_{K,\widehat{\xi}^k}(\mathbf{x}^{(j)}) - P_{K,\mu}(\mathbf{x}^{(j)}) \right] = 0$ for all j with $\mathbf{x}^{(j)} \in \mathrm{supp}(\widehat{\xi}^k)$, with $\widetilde{\mathbf{g}}_k$ the gradient of $\widetilde{g}(\cdot)$ at $\widetilde{\omega}_k$; see the proof of Theorem 4-(*iii*). Therefore, $P_{K,\widehat{\xi}^k}(\mathbf{x}^{(j)}) = P_{K,\mu}(\mathbf{x}^{(j)})$ for all $\mathbf{x}^{(j)} \in \mathrm{supp}(\widetilde{\xi}^k)$; Algorithm 3-(*iii*) can be stopped when $P_{K,\widehat{\xi}^k}(\mathbf{x}) \geq P_{K,\mu}(\mathbf{x})$ for all $\mathbf{x} \in \mathscr{X}_C$ and we can add the following line to Algorithm 3-(*iii*):

4'-(*iii*): if $S_{k-1}(\mathbf{x}_k) - P_{K,\mu}(\mathbf{x}_k) \geq 0$ **then return** $\mathbf{X}_{k-1}$, ξ_{k-1} and stop;

Proof of Theorem 5. We have $\mathscr{E}_K(\nu - \mu) = \mathscr{E}_{K_\mu}(\nu)$ for any $\nu \in \mathscr{M}_{[1]}(\mathscr{X}_C)$ with K_μ the reduced kernel defined by (10); see Damelin et al. (2010), Pronzato and Zhigljavsky (2020, Th. 3.5). Let ω_k be the vector of weights associated with ξ_k at step k. Since $\mathsf{MMD}_K^2(\mu, \xi_k) = \omega_k^\top \mathbf{K}_{\mu_C} \omega_k$, see (11), we have

$$(k+1)^2\,\mathsf{MMD}_K^2(\mu, \xi_{k+1}) = k^2\,\mathsf{MMD}_K^2(\mu, \xi_k) + 2k\,\omega_k^\top \mathbf{K}_{\mu_C} \mathbf{e}_{j_{k+1}} + K_\mu(\mathbf{x}^{(j_{k+1})}, \mathbf{x}^{(j_{k+1})}),$$

where $j_{k+1} \in \mathrm{Arg\,min}_{j \in \mathbb{I}_C} 2k\,\omega_k^\top \mathbf{K}_{\mu_C} \mathbf{e}_j + K_\mu(\mathbf{x}^{(j)}, \mathbf{x}^{(j)})$, and $\mathbf{x}^{(j_{k+1})}$ coincides with (14) when $\mathscr{X}_C$ is substituted for $\mathscr{X}$. Therefore, for any $\omega \in \mathscr{P}_C$, we have

$$\begin{aligned}(k+1)^2\,\mathsf{MMD}_K^2(\mu, \xi_{k+1}) &\leq k^2\,\mathsf{MMD}_K^2(\mu, \xi_k) + 2k\,\omega_k^\top \mathbf{K}_{\mu_C} \omega + \overline{K}_{\mu,C} \\ &\leq k^2\,\mathsf{MMD}_K^2(\mu, \xi_k) + 2k\,(\omega_k^\top \mathbf{K}_{\mu_C} \omega_k)^{1/2} (\omega^\top \mathbf{K}_{\mu_C} \omega)^{1/2} + \overline{K}_{\mu,C}, \end{aligned} \tag{53}$$

where $\overline{K}_{\mu,C} = \max_{\mathbf{x} \in \mathscr{X}_C} K_\mu(\mathbf{x}, \mathbf{x}) \leq A_C$, see (17). The inequality (53) is satisfied in particular for $\omega_*^C \in \mathrm{Arg\,min}_{\omega \in \mathscr{P}_C} \omega^\top \mathbf{K}_{\mu_C} \omega$, with ${\omega_*^C}^\top \mathbf{K}_{\mu_C} \omega_*^C = M_C^2$, therefore

$$(k+1)^2\,\mathsf{MMD}_K^2(\mu, \xi_{k+1}) \leq k^2\,\mathsf{MMD}_K^2(\mu, \xi_k) + 2k\,M_C\,\mathsf{MMD}_K(\mu, \xi_k) + A_C\,. \tag{54}$$

We prove (26) by induction on n. For $n = 1$, $\mathsf{MMD}_K^2(\mu, \delta_{\mathbf{x}_1}) = K_\mu(\mathbf{x}_1, \mathbf{x}_1) \leq \overline{K}_{\mu,C} \leq A_C$. Suppose that (26) is true for n. Then, (54) implies

$$\begin{aligned}\mathsf{MMD}_K^2(\mu, \xi_{n+1}) &\leq \frac{1}{(n+1)^2} \left\{ n^2 \left(M_C^2 + A_C\,\frac{1+\log n}{n} \right) + 2n\,M_C \left(M_C^2 + A_C\,\frac{1+\log n}{n} \right)^{1/2} + A_C \right\} \\ &\leq \frac{1}{(n+1)^2} \left\{ n^2 \left(M_C^2 + A_C\,\frac{1+\log n}{n} \right) + n \left(2\,M_C^2 + A_C\,\frac{1+\log n}{n} \right) + A_C \right\} \\ &= M_C^2 + A_C\,\frac{1+\log(n+1)}{n+1} - \frac{M_C^2 + A_C\,[(n+1)\log(1+1/n) - 1]}{(n+1)^2} \\ &= M_C^2 + A_C\,\frac{1+\log(n+1)}{n+1} - \frac{M_C^2/(n+1) + A_C\,[-\log(1-1/(n+1)) - 1/(n+1)]}{n+1} \\ &\leq M_C^2 + A_C\,\frac{1+\log(n+1)}{n+1} \end{aligned}$$

since $\log(1+x) \leq x$ for $x > -1$, which concludes the proof of (26). ∎

Proof of Theorem 6. Using (38) and (40), we get

$$\min_{\mathbf{x}\in\mathscr{X}_C} \Delta_C[\xi_k^+(\mathbf{x},\alpha_{k+1})] \leq \widehat{g}_C(\omega_k) - \widehat{g}_C(\omega_*^C) + \min_{j\in\mathbb{I}_C} 2\,\alpha_{k+1}\,(\mathbf{e}_j - \omega_k)^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C) + B_C\,\alpha_{k+1}^2\,,$$

and, from the same arguments as in the proof of Theorem 1,

$$\Delta_C(\xi_{k+1}) = \min_{\mathbf{x}\in\mathscr{X}_C} \Delta_C[\xi_k^+(\mathbf{x},\alpha_{k+1})] \leq (1-\alpha_{k+1})\,\Delta_C(\xi_k) + B_C\,\alpha_{k+1}^2\,. \tag{55}$$

When $\alpha_k = 2/(k+1)$ for all k, Lemma 2-(*ii*) gives (22).

Similarly, when $\widehat{\xi}^C = \xi_*^C$, we have

$$\Delta_C(\xi_{k+1}) = \min_{\mathbf{x}\in\mathscr{X}_C} \Delta_C[\xi_k^+(\mathbf{x},\alpha_{k+1})] \leq (1-2\,\alpha_{k+1})\,\Delta_C(\xi_k) + B_C\,\alpha_{k+1}^2\,,$$

see the proof of Theorem 1, and Lemma 2-(*iii*) gives (21). ∎

Proof of Theorem 7. For $\alpha \in [0,1]$, any $\mathbf{x}^{(j)} \in \mathscr{X}_C$ satisfies

$$\Delta_C[\xi_k^+(\mathbf{x}^{(j)},\alpha)] = \widehat{g}_C(\omega_k) - \widehat{g}(\omega_*^C) + 2\,\alpha\,(\mathbf{e}_j - \omega_k)^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C) + \alpha^2\,\|\mathbf{e}_j - \omega_k\|_{\mathbf{K}_C}^2\,, \tag{56}$$

see (38), the right-hand side of which is minimum when

$$\alpha = \widehat{\alpha}_{k+1,j} = \frac{(\mathbf{e}_j - \omega_k)^\top \mathbf{K}_C(\widehat{\omega}^C - \omega_k)}{\|\mathbf{e}_j - \omega_k\|_{\mathbf{K}_C}^2}\,.$$

By restricting α to $[0,1]$, we obtain $[\mathbf{x}_{k+1},\alpha_{k+1}] = [\mathbf{x}^{(j_{k+1}^*)},\alpha_{k+1,j}^*]$ with

$$\begin{aligned}
\alpha_{k+1,j}^* &= \max\{0,\min\{\widehat{\alpha}_{k+1,j},1\}\} \\
j_{k+1}^* &= \arg\min_{j\in\mathbb{I}_C} 2\,\alpha_{k+1,j}^*\,(\mathbf{e}_j - \omega_k)^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C) + (\alpha_{k+1,j}^*)^2\,\|\mathbf{e}_j - \omega_k\|_{\mathbf{K}_C}^2\,.
\end{aligned}$$

Using (46) and (47), we obtain that $\widehat{\alpha}_{k+1,j}$ is given by (23) with $\mathbf{x}^{(j)}$ substituted for $\mathbf{x}_{k+1}$. The recursive updating of $Q_k = \sum_{i,\ell=1}^k \{\mathbf{w}_k\}_i \{\mathbf{w}_k\}_\ell K(\mathbf{x}_i,\mathbf{x}_\ell)$, $R_k = \sum_{i=1}^k \{\mathbf{w}_k\}_i P_{K,\mu}(\mathbf{x}_i)$ and $S_k(\mathbf{x}) = P_{K,\xi_k}(\mathbf{x})$ gives Algorithm 5. As in the proof of Theorem 3, $\mathsf{MMD}_K(\mu,\xi_k) > \mathsf{MMD}_K(\mu,\xi_*^C)$ implies that there exists $j \in \mathbb{I}_C$ such that $(\mathbf{e}_j - \omega_k)^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C) < 0$, and therefore $\widehat{\alpha}_{k+1,j} = \alpha(\mathbf{x}^{(j)}) > 0$. Conversely, $\alpha(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathscr{X}_C$ implies that $\mathsf{MMD}_K(\mu,\xi_k) = \mathsf{MMD}_K(\mu,\xi_*^C)$ and Algorithm 5 can be stopped.

As $\|\mathbf{e}_j - \omega_k\|_{\mathbf{K}_C}^2 \leq B_C$ in (56), see (40), we have

$$\min_{\mathbf{x}\in\mathscr{X}_C,\alpha\in[0,1]} \Delta_C[\xi_k^+(\mathbf{x},\alpha)] \leq \widehat{g}_C(\omega_k) - \widehat{g}_C(\omega_*^C) + \min_{j\in\mathbb{I}_C} 2\,\alpha_{k+1}\,(\mathbf{e}_j - \omega_k)^\top \mathbf{K}_C(\omega_k - \widehat{\omega}^C) + B_C\,\alpha_{k+1}^2$$

for any predefined choice of α_{k+1} in $[0,1]$. The rest of the proof is similar to that of Theorem 3. ∎

Proof of Theorem 8. For any ξ_k, the value of $\Delta_C(\xi_{k+1})$ obtained by SBQ cannot exceed that obtained by IWO applied to KH, which yields the bounds given in Theorem 8 for (9) and (13).

Denote $\xi^{++}(\mathbf{x},\alpha) = \xi + w\delta_{\mathbf{x}}$ for any $\xi \in \mathscr{M}(\mathscr{X}_C)$, $\mathbf{x} \in \mathscr{X}_C$ and $w \in \mathbb{R}$, so that (27) corresponds to $\xi_{k+1} = \xi_k^{++}(\mathbf{x}_{k+1},w_{k+1})$ with $[\mathbf{x}_{k+1},w_{k+1}] \in \operatorname{Arg\,min}_{\mathbf{x}\in\mathscr{X}_C,\,w} \mathsf{MMD}_K^2[\xi_k^{++}(\mathbf{x},w)]$. We get

$$\Delta_C(\xi_{k+1}) = \Delta_C(\xi_k) - \frac{[\mathbf{e}_{j_{k+1}}^\top \mathbf{K}_C(\omega_k - \widetilde{\omega}^C)]^2}{K(\mathbf{x}_{k+1},\mathbf{x}_{k+1})}\,,$$

where $j_{k+1} \in \operatorname{Arg\,max}_{j\in\mathbb{I}_C} [\mathbf{e}_j^\top \mathbf{K}_C(\omega_k - \widetilde{\omega}^C)]^2/K(\mathbf{x}^{(j)},\mathbf{x}^{(j)})$, see the proof of Theorem 4-(*iii*). The convexity of $\widetilde{g}(\cdot)$ implies $\widetilde{g}_C(\omega_*^C) \geq \widetilde{g}_C(\omega_k) + 2\,{\omega_*^C}^\top \mathbf{K}_C(\omega_k - \widetilde{\omega}^C)$, see (50). Therefore, as long as

$\Delta_C(\xi_k) = \widetilde{g}_C(\omega_k) - \widetilde{g}_C(\omega_*^C) \geq 0$, $[{\omega_*^C}^\top \mathbf{K}_C(\omega_k - \widetilde{\omega}^C)]^2 \geq [\widetilde{g}_C(\omega_k) - \widetilde{g}_C(\omega_*^C)]^2/4$. The maximum of $[\omega^\top \mathbf{K}_C(\omega_k - \widetilde{\omega}^C)]^2$ with respect to $\omega \in \mathscr{P}_C$ is attained on a vertex of $\mathscr{P}_C$, which implies that $[\mathbf{e}_{j_{k+1}}^\top \mathbf{K}_C(\omega_k - \widetilde{\omega}^C)]^2 \geq [\widetilde{g}_C(\omega_k) - \widetilde{g}_C(\omega_*^C)]^2/4$ when $K(\mathbf{x},\mathbf{x}) = \overline{K}_C$ for all $\mathbf{x}$, and $\Delta_C(\xi_k)$ satisfies (52). The conclusion is the same as for Theorem 4-(*iii*). ■

Proof of Lemma 1. A_C depends on $\mathscr{X}_C$, but $A_C \leq A(\mu)$ since $\overline{K}_C \leq \overline{K}$; similarly, $B_C \leq B$. Since $M_C^2 \leq \mathbf{1}_C^\top \mathbf{K}_{\mu_C} \mathbf{1}_C / C^2$, we get

$$\begin{aligned}
\mathsf{E}\{M_C^2\} &\leq \frac{1}{C^2}\,\mathsf{E}\left\{\sum_{i,j=1}^{C} K_\mu(\mathbf{x}^{(i)},\mathbf{x}^{(j)})\right\} \\
&= \frac{1}{C^2}\,\mathsf{E}\left\{\sum_{i,j=1}^{C} K(\mathbf{x}^{(i)},\mathbf{x}^{(j)})\right\} - \frac{2}{C}\,\mathsf{E}\left\{\sum_{i=1}^{C} P_{K,\mu}(\mathbf{x}^{(i)})\right\} + \mathscr{E}_K(\mu) \\
&= \frac{1}{C^2}\,\mathsf{E}\left\{\sum_{i=1}^{C} K(\mathbf{x}^{(i)},\mathbf{x}^{(i)})\right\} + \frac{1}{C^2}\,\mathsf{E}\left\{\sum_{\substack{i,\,j\,=\,1\\ i\,\neq\,j}}^{C} K(\mathbf{x}^{(i)},\mathbf{x}^{(j)})\right\} - \mathscr{E}_K(\mu) \\
&= \frac{\tau_1(\mu)}{C} + \frac{C(C-1)}{C^2}\,\mathscr{E}_K(\mu) - \mathscr{E}_K(\mu) = \frac{\tau_1(\mu) - \mathscr{E}_K(\mu)}{C}\,.
\end{aligned}$$

■

Proof of Theorem 9. We have $\mathsf{MMD}_K^2(\mu,\xi_{n,e}) = \mathbf{1}_n^\top \mathbf{K}_{\mu_n} \mathbf{1}_n / n^2$, with $\mathsf{E}_\mu\{K_\mu(\cdot,X)\} \equiv 0$ on $\mathscr{X}$ and $\int_{\mathscr{X}^2} K_\mu(\mathbf{x},\mathbf{x}')\,\mathrm{d}\mu(\mathbf{x})\mathrm{d}\mu(\mathbf{x}') = \mathscr{E}_{K_\mu}(\mu) = 0$. From Serfling (1980, p. 194), the U-statistic $U_n = 2/[n(n-1)] \sum_{i<j} K_\mu(\mathbf{x}_i,\mathbf{x}_j)$ satisfies $n\,U_n \xrightarrow{\mathrm{d}} Y = \sum_{i=1}^\infty \lambda_i(\chi_{1i}^2 - 1)$. The V-statistic $V_n = (1/n^2) \sum_{i,j} K_\mu(\mathbf{x}_i,\mathbf{x}_j) = \mathsf{MMD}_K^2(\mu,\xi_{n,e})$ satisfies $V_n = (1-1/n)\,U_n + (1/n^2) \sum_i K_\mu(\mathbf{x}_i,\mathbf{x}_i)$, with $U_n \xrightarrow{\mathrm{a.s.}} 0$ and $(1/n) \sum_i K_\mu(\mathbf{x}_i,\mathbf{x}_i) \xrightarrow{\mathrm{a.s.}} \sum_{i=1}^\infty \lambda_i$. Therefore, $n\,V_n \xrightarrow{\mathrm{d}} Z = \sum_{i=1}^\infty \lambda_i \chi_{1i}^2$. ■

Appendix C: multiple random candidate sets

Suppose that $\mathbf{x}_{k+1}$ is selected within $\mathscr{X}_C[k+1]$ at iteration k. Denote $\mathscr{X}_{C_{k+1}} = \cup_{i=1}^{k+1} \mathscr{X}_C[i]$ and let $\mathscr{P}_{C_{k+1}}$ be the corresponding probability simplex in $\mathbb{R}^{C_{k+1}}$ (with $C_{k+1} = (k+1)\,C$ when the $\mathscr{X}_C[i]$ do not intersect). To a measure ξ_k in $\mathscr{M}(\mathscr{X}_{C_k})$ (respectively, in $\mathscr{M}_{[1]}^+(\mathscr{X}_{C_k})$) corresponds a vector of weights ω_k in $\mathbb{R}^{C_k}$ (respectively, in $\mathscr{P}_{C_k}$), and we denote by ω_k' the same vector plunged into $\mathbb{R}^{C_{k+1}}$ (respectively, into $\mathscr{P}_{C_{k+1}}$); $\xi_*^{C[k+1]}$ is the measure in $\mathscr{M}_{[1]}^+(\mathscr{X}_C[k+1])$ that minimises $\mathsf{MMD}(\mu,\xi)$ and $\omega_*^{C[k+1]'}$ is the vector of associated weights in $\mathscr{P}_{C_{k+1}}$. Similarly, $\widehat{\omega}^{C_{k+1}}$ denotes the vector of weights for the optimal measure in $\mathscr{M}_{[1]}^+(\mathscr{X}_{C_{k+1}})$.

Consider one-step-ahead algorithms (Algorithms 1, 2, 4 and 5) and $\xi_k^+[\mathbf{x}_{k+1},\alpha_{k+1}] = (1-\alpha_{k+1})\,\xi_k + \alpha_{k+1}\,\delta_{\mathbf{x}_{k+1}}$ constructed at iteration k. We have

$$\begin{aligned}
\mathsf{MMD}_K^2(\mu,\xi_k^+[\mathbf{x}_{k+1},\alpha_{k+1}]) - \mathsf{MMD}_K^2(\mu,\xi_*^{C[k+1]}) &= \widehat{g}_{C_{k+1}}(\omega_k') - \widehat{g}_{C_{k+1}}(\omega_*^{C[k+1]'}) \\
&\quad + 2\,\alpha_{k+1}\,(\mathbf{e}_{j_{k+1}} - \omega_k')^\top \mathbf{K}_{C_{k+1}}(\omega_k' - \widehat{\omega}^{C_{k+1}}) + \alpha_{k+1}^2 \|\mathbf{e}_{j_{k+1}} - \omega_k'\|_{\mathbf{K}_{C_{k+1}}}^2\,,
\end{aligned}$$

with $\mathbf{e}_{j_{k+1}}$ the basis vector in $\mathbb{R}^{C_{k+1}}$ corresponding to $\mathbf{x}_{k+1} \in \mathscr{X}_C[k+1]$, and $\|\mathbf{e}_{j_{k+1}} - \omega_k'\|_{\mathbf{K}_{C_{k+1}}}^2 \leq B$, see (38), (40) and Lemma 1. The convexity of $\widehat{g}_{C_{k+1}}(\cdot)$ implies

$$\widehat{g}_{C_{k+1}}(\omega_*^{C[k+1]'}) \geq \widehat{g}_{C_{k+1}}(\omega_k') + 2\,(\omega_*^{C[k+1]'} - \omega_k')^\top \mathbf{K}_{C_{k+1}}(\omega_k' - \widehat{\omega}^{C_{k+1}})$$

so that $2\,(\mathbf{e}_{j_{k+1}} - \omega'_k)^\top \mathbf{K}_{C_{k+1}}(\omega'_k - \widehat{\omega}^{C_{k+1}}) \leq \widehat{g}_{C_{k+1}}(\omega_*^{C[k+1]'}) - \widehat{g}_{C_{k+1}}(\omega'_k)$ when $\mathbf{x}_{k+1}$ is chosen by kernel herding. This gives

$$\begin{aligned}\mathsf{MMD}_K^2(\mu, \xi_k^+[\mathbf{x}_{k+1}, \alpha_{k+1}]) - \mathsf{MMD}_K^2(\mu, \xi_*^{C[k+1]}) \quad &\leq \\ (1-\alpha_{k+1}) \, \left[\mathsf{MMD}_K^2(\mu, \xi_k) - \mathsf{MMD}_K^2(\mu, \xi_*^{C[k+1]})\right] &+ B\,\alpha_{k+1}^2\,.\end{aligned}$$

When each $\mathscr{X}_C[i]$ is made of independent samples from μ, we have $\mathsf{E}_\mu\{\mathsf{MMD}_K^2(\mu, \xi_*^{C[i]})\} = \mathsf{E}_\mu\{M_C^2\}$ for all i, and therefore, when α_{k+1} is predefined (deterministic),

$$\mathsf{E}_\mu\{\Delta(\xi_{k+1})\} \leq (1-\alpha_{k+1})\,\mathsf{E}_\mu\{\Delta(\xi_k)\} + B\,\alpha_{k+1}^2\,,$$

where we have denoted $\Delta(\xi) = \Delta_{C[1]}(\xi) = \mathsf{MMD}_K^2(\mu, \xi) - \mathsf{MMD}_K^2(\mu, \xi_*^{C[1]})$. From the same arguments as those used in the proof of Theorem 1, we thus obtain that the measure generated by Algorithm 1 with $\alpha_k = 1/k$ and using a set $\mathscr{X}_C[k]$ composed of C independent samples from μ for all k satisfies $\mathsf{E}_\mu\{\mathsf{MMD}_K^2(\mu, \xi_k)\} \leq \mathsf{E}_\mu\{M_C^2\} + B\,(2 + \log n)/(n+1)$, $n \geq 1$, compare with (20). When $\alpha_k = 2/(k+1)$ in Algorithm 1, then $\mathsf{E}_\mu\{\mathsf{MMD}_K^2(\mu, \xi_k)\} \leq \mathsf{E}_\mu\{M_C^2\} + 4\,B/(n+3)$, $n \geq 1$, compare with (22). Following the arguments used in the proof of Theorem 3, see (44), we get the same bound for Algorithm 2. Also, following the arguments in the proofs of Theorems 6 and 7, similar bounds are obtained for Algorithms 4 and 5, which extends the results in those theorems to this situation of multiple random candidate sets. A similar extension applies to Algorithm 3-(i); see the proof of Theorem 4-(i).

Consider now Algorithm 3-(ii) and (iii) and SBQ. For Algorithm 3-(ii), we have

$$\mathsf{MMD}_K^2(\mu, \xi_{k+1}) \leq \mathsf{MMD}_K^2(\mu, \xi_k) - \frac{\left[(\mathbf{e}_{j_{k+1}} - \omega'_k)^\top \mathbf{K}_{C_{k+1}}(\omega'_k - \widehat{\omega}^{C_{k+1}})\right]^2}{B}$$

and the convexity of $\widehat{g}_{C_{k+1}}(\cdot)$ gives

$$\mathsf{MMD}_K^2(\mu, \xi_{k+1}) \leq \mathsf{MMD}_K^2(\mu, \xi_k) - \frac{\left[\widehat{g}_{C_{k+1}}(\omega'_k) - \widehat{g}_{C_{k+1}}(\omega_*^{C[k+1]'})\right]^2}{4\,B}\,.$$

Since $\mathsf{E}_\mu\{\mathsf{MMD}_K^2(\mu, \xi_*^{C[i]})\} = \mathsf{E}_\mu\{M_C^2\}$ for all i, Jensen's inequality gives

$$\mathsf{E}_\mu\left\{\left[\widehat{g}_{C_{k+1}}(\omega'_k) - \widehat{g}_{C_{k+1}}(\omega_*^{C[k+1]'})\right]^2\right\} \quad \geq \quad \left[\mathsf{E}_\mu\left\{\widehat{g}_{C_{k+1}}(\omega'_k) - \widehat{g}_{C_{k+1}}(\omega_*^{C[k+1]'})\right\}\right]^2 = \mathsf{E}_\mu^2\{\Delta(\xi_k)\}\,.$$

We thus obtain

$$\mathsf{E}_\mu\{\Delta(\xi_{k+1})\} \leq \mathsf{E}_\mu\{\Delta(\xi_k)\} - \frac{\mathsf{E}_\mu^2\{\Delta(\xi_k)\}}{4\,B}\,,$$

an inequality similar to (49) in the proof of Theorem 4-(ii), and ξ_n satisfies an inequality of the form (22) where each term is replaced by its expected value.

Similar developments yield an inequality similar to (52), with expected values everywhere, and thus an extension of (25) for Algorithm 3-(iii). Theorem 8, that indicates that the performance of SBQ cannot be worse that that of Algorithm 3, continues to apply; the details are omitted.

Acknowledgments

This work was partly supported by project INDEX (INcremental Design of EXperiments) ANR-18-CE91-0007 of the French National Research Agency (ANR). The author would like to thank Chris Oates for sending him a preprint of the paper (Teymur et al., 2021) which was deeply inspiring.

The author is grateful to the two anonymous referees for their careful reading and their comments and suggestions which helped to improve the paper.

References

Ahipaşaoğlu, S., Sun, P., and Todd, M. (2008). Linear convergence of a modified Frank-Wolfe algorithm for computing minimum-volume enclosing ellipsoids. *Optimization Mehods and Software*, 23:5–19.

Atwood, C. (1973). Sequences converging to D-optimal designs of experiments. *Annals of Statistics*, 1(2):342–352.

Bach, F., Lacoste-Julien, S., and Obozinski, G. (2012). On the equivalence between herding and conditional gradient algorithms. In *Proc. 29th Annual International Conference on Machine Learning*, pages 1355–1362.

Briol, F.-X., Oates, C., Girolami, M., and Osborne, M. (2015). Frank-Wolfe Bayesian quadrature: Probabilistic integration with theoretical guarantees. In *Advances in Neural Information Processing Systems*, pages 1162–1170.

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

Chen, W., Barp, A., Briol, F.-X., Gorham, J., Girolami, M., Mackey, L., and Oates, C. (2019). Stein point Markov Chain Monte Carlo. *arXiv preprint arXiv:1905.03673*.

Chen, W., Mackey, L., Gorham, J., Briol, F.-X., and Oates, C. (2018). Stein points. *arXiv preprint arXiv:1803.10161v4, Proc. ICML*.

Chen, Y., Welling, M., and Smola, A. (2010). Super-samples from kernel herding. In *Proceedings 26th Conference on Uncertainty in Artificial Intelligence (UAI'10)*, pages 109–116, Catalina Island, CA. AUAI Press Arlington, Virginia. arXiv preprint arXiv:1203.3472.

Clarkson, K. (2010). Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4):63.

Damelin, S., Hickernell, F., Ragozin, D., and Zeng, X. (2010). On energy, discrepancy and group invariant measures on measurable subsets of Euclidean space. *J. Fourier Anal. Appl.*, 16:813–839.

Detommaso, G., Cui, T., Marzouk, Y., Spantini, A., and Scheichl, R. (2018). A Stein variational Newton method. In *Advances in Neural Information Processing Systems*, pages 9187–9197.

Dunn, J. (1980). Convergence rates for conditional gradient sequences generated by implicit step length rules. *SIAM J. Control and Optimization*, 18(5):473–487.

Dunn, J. and Harshbarger, S. (1978). Conditional gradient algorithms with open loop step size rules. *Journal of Mathematical Analysis and Applications*, 62:432–444.

Fang, K.-T., Li, R., and Sudjianto, A. (2006). *Design and Modeling for Computer Experiments*. Chapman & Hall/CRC, Boca Raton.

Fedorov, V. (1972). *Theory of Optimal Experiments*. Academic Press, New York.

Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Res. Logist. Quart.*, 3:95–110.

Garreau, D., Jitkrittum, W., and Kanagawa, M. (2017). Large sample analysis of the median heuristic. *arXiv preprint arXiv:1707.07269*.

Gorham, J. and MacKey, L. (2017). Measuring sample quality with kernels. *arXiv preprint arXiv:1703.01717*.

Graf, S. and Luschgy, H. (2000). *Foundations of Quantization for Probability Distributions*. Springer, Berlin.

Hickernell, F. (1998). A generalized discrepancy and quadrature error bound. *Mathematics of Computation*, 67(221):299–322.

Huszár, F. and Duvenaud, D. (2012). Optimally-weighted herding is Bayesian quadrature. In *Proceedings 28th Conference on Uncertainty in Artificial Intelligence (UAI'12)*, pages 377–385, Catalina Island, CA. AUAI Press Arlington, Virginia. arXiv preprint arXiv:1204.1664.

Joseph, V., Dasgupta, T., Tuo, R., and Wu, C. (2015a). Sequential exploration of complex surfaces using minimum energy designs. *Technometrics*, 57(1):64–74.

Joseph, V., Gul, E., and Ba, S. (2015b). Maximum projection designs for computer experiments. *Biometrika*, 102(2):371–380.

Joseph, V., Wang, D., Gu, L., Lyu, S., and Tuo, R. (2019). Deterministic sampling of expensive posteriors using minimum energy designs. *Technometrics*, 61(3):297–308.

Karvonen, T., Kanagawa, M., and Särkkä, S. (2019). On the positivity and magnitudes of Bayesian quadrature weights. *Statistics and Computing*, 29(6):1317–1333.

Lacoste-Julien, S. and Jaggi, M. (2015). On the global linear convergence of Frank-Wolfe optimization variants. *Advances in Neural Processing Information Systems*, 28:496–504. arXiv preprint arXiv:1511.05932v1.

Liu, Q. and Wang, D. (2016). Stein variational gradient descent: a general purpose Bayesian inference algorithm. *Advances In Neural Information Processing Systems*, pages 2378–2386. arXiv preprint arXiv:1608.04471v2.

Mak, S. and Joseph, V. (2017). Projected support points, with application to optimal MCMC reduction. *arXiv preprint arXiv:1708.06897*.

Mak, S. and Joseph, V. (2018). Support points. *Annals of Statistics*, 46(6A):2562–2592.

Oates, C., Girolami, M., and Chopin, N. (2017). Control functionals for Monte Carlo integration. *Journal of Royal Statistical Society*, B79(3):695–718.

Pronzato, L. (2017). Minimax and maximin space-filling designs: some properties and methods for construction. *Journal de la Société Française de Statistique*, 158(1):7–36.

Pronzato, L. and Müller, W. (2012). Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22:681–701.

Pronzato, L. and Pázman, A. (2013). *Design of Experiments in Nonlinear Models. Asymptotic Normality, Optimality Criteria and Small-Sample Properties*. Springer, LNS 212, New York.

Pronzato, L. and Zhigljavsky, A. (2020). Bayesian quadrature, energy minimization and space-filling design. *SIAM/ASA J. Uncertainty Quantification*, 8(3):959–1011.

Pronzato, L. and Zhigljavsky, A. (2021). Minimum-energy measures for singular kernels. *Journal of Computational and Applied Mathematics*, 382. (113089, 16 pages) hal-02495643.

Sejdinovic, S., Sriperumbudur, B., Gretton, A., and Fukumizu, K. (2013). Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5):2263–2291.

Serfling, R. (1980). *Approximation Theorems of Mathematical Statistics*. Wiley, New York.

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

Székely, G. and Rizzo, M. (2013). Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8):1249–1272.

Teymur, O., Gorham, J., Riabiz, M., and Oates, C. (2021). Optimal quantisation of probability measures using maximum mean discrepancy. In *International Conference on Artificial Intelligence and Statistics*, pages 1027–1035. arXiv preprint arXiv:2010.07064v1.

Todd, M. and Yildirim, E. (2007). On Khachiyan's algorithm for the computation of minimum volume enclosing ellipsoids. *Discrete Applied Math.*, 155:1731–1744.

Wolfe, P. (1970). Convergence theory in nonlinear programming. In Abadie, J., editor, *Integer and Nonlinear Programming*, pages 1–36. North-Holland, Amsterdam.

Wolfe, P. (1976). Finding the nearest point in a polytope. *Mathematical Programming*, 11:128–149.

Wright, S. (2015). Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34.

Wynn, H. (1970). The sequential generation of D-optimum experimental designs. *Annals of Math. Stat.*, 41:1655–1664.

Zhigljavsky, A., Pronzato, L., and Bukina, E. (2012). An asymptotically optimal gradient algorithm for quadratic optimization with low computational cost. *Optimization Letters*. DOI 10.1007/s11590-012-0491-7.