AUTOMATED SCALABLE BAYESIAN INFERENCE VIA HILBERT CORESETS

TREVOR CAMPBELL AND TAMARA BRODERICK

Abstract. The automation of posterior inference in Bayesian data analysis has enabled experts and nonexperts alike to use more sophisticated models, engage in faster exploratory modeling and analysis, and ensure experimental reproducibility. However, standard automated posterior inference algorithms are not tractable at the scale of massive modern datasets, and modifications to make them so are typically model-specific, require expert tuning, and can break theoretical guarantees on inferential quality. Building on the Bayesian coresets framework, this work instead takes advantage of data redundancy to shrink the dataset itself as a preprocessing step, providing fully-automated, scalable Bayesian inference with theoretical guarantees. We begin with an intuitive reformulation of Bayesian coreset construction as sparse vector sum approximation, and demonstrate that its automation and performance-based shortcomings arise from the use of the supremum norm. To address these shortcomings we develop Hilbert coresets, i.e., Bayesian coresets constructed under a norm induced by an inner-product on the log-likelihood function space. We propose two Hilbert coreset construction algorithms—one based on importance sampling, and one based on the Frank-Wolfe algorithm—along with theoretical guarantees on approximation quality as a function of coreset size. Since the exact computation of the proposed inner-products is model-specific, we automate the construction with a random finite-dimensional projection of the log-likelihood functions. The resulting automated coreset construction algorithm is simple to implement, and experiments on a variety of models with real and synthetic datasets show that it provides high-quality posterior approximations and a significant reduction in the computational cost of inference.

1. Introduction

Bayesian probabilistic models are a standard tool of choice in modern data analysis. Their rich hierarchies enable intelligent sharing of information across subpopulations, their posterior distributions provide many avenues for principled parameter estimation and uncertainty quantification, and they can incorporate expert knowledge through the prior. In all but the simplest models, however, the posterior distribution is intractable to compute exactly, and we must resort to approximate inference algorithms. Markov chain Monte Carlo (MCMC) (Gelman et al., 2013, Chapters 11, 12) methods are the gold standard, due primarily to their guaranteed asymptotic exactness. Variational Bayes (VB) (Jordan et al., 1999; Wainwright and Jordan, 2008) is also becoming widely used due to its tractability, detectable convergence, and parameter estimation performance in practice.

One of the most important recent developments in the Bayesian paradigm has been the automation of these standard inference algorithms. Rather than having to develop, code, and tune specific instantiations of MCMC or VB for each model,

practitioners now have "black-box" implementations that require only a basic specification of the model as inputs. For example, while standard VB requires the specification of model gradients—whose formulae are often onerous to obtain—and an approximating family—whose rigorous selection is an open question—ADVI (Ranganath et al., 2014; Kucukelbir et al., 2015, 2017) applies standard transformations to the model so that a multivariate Gaussian approximation can be used, and computes gradients with automatic differentiation. The user is then left only with the much simpler task of specifying the log-likelihood and prior. Similarly, while Hamiltonian Monte Carlo (Neal, 2011) requires tuning a step size and path length parameter, NUTS (Hoffman and Gelman, 2014) provides a method for automatically determining reasonable values for both. This level of automation has many benefits: it enables experts and nonexperts alike to use more sophisticated models, it facilitates faster exploratory modeling and analysis, and helps ensure experimental reproducibility.

But as modern datasets continue to grow larger over time, it is important for inference to be not only automated, but *scalable* while retaining *theoretical guarantees* on the quality of inferential results. In this regard, the current set of available inference algorithms falls short. Standard MCMC algorithms may be "exact", but they are typically not tractable for large-scale data, as their complexity per posterior sample scales at least linearly in the dataset size. Variational methods on the other hand are often scalable, but posterior approximation guarantees continue to elude researchers in all but a few simple cases. Other scalable Bayesian inference algorithms have largely been developed by modifying standard inference algorithms to handle distributed or streaming data processing. Examples include subsampling and streaming methods for variational Bayes (Hoffman et al., 2013; Broderick et al., 2013; Campbell et al., 2015), subsampling methods for MCMC (Welling and Teh, 2011; Ahn et al., 2012; Bardenet et al., 2014; Korattikara et al., 2014; Maclaurin and Adams, 2014; Bardenet et al., 2015), and distributed "consensus" methods for MCMC (Scott et al., 2016; Srivastava et al., 2015; Rabinovich et al., 2015; Entezari et al., 2016). These methods either have no guarantees on the quality of their inferential results, or require expensive iterative access to a constant fraction of the data, but more importantly they tend to be model-specific and require extensive expert tuning. This makes them poor candidates for automation on the large class of models to which standard automated inference algorithms are applicable.

An alternative approach, based on the observation that large datasets often contain redundant data, is to modify the dataset itself such that its size is reduced while preserving its original statistical properties. In Bayesian regression, for example, a large dataset can be compressed using random linear projection (Geppert et al., 2017; Ahfock et al., 2017; Bardenet and Maillard, 2015). For a wider class of Bayesian models, one can construct a small weighted subset of the data, known as a *Bayesian coreset*[1] (Huggins et al., 2016), whose weighted log-likelihood approximates the full data log-likelihood. The coreset can then be passed to a standard (automated) inference algorithm, providing posterior inference at a significantly reduced computational cost.

[1]The concept of a coreset originated in computational geometry and optimization (Agarwal et al., 2005; Feldman and Langberg, 2011; Feldman et al., 2013; Bachem et al., 2015; Lucic et al., 2016; Bachem et al., 2016; Feldman et al., 2011; Han et al., 2016).

Bayesian coresets, in contrast to other large-scale inference techniques, are simple to implement, computationally inexpensive, and have theoretical guarantees relating coreset size to both computational complexity and the quality of approximation (Huggins et al., 2016). However, their construction cannot be easily automated, as it requires computing the *sensitivity* (Langberg and Schulman, 2010) of each data point, a model-specific task that involves significant technical expertise. This approach also often necessitates a bounded parameter space to ensure bounded sensitivities, precluding many oft-used continuous likelihoods and priors. Further, since Bayesian coreset construction involves i.i.d. random subsampling, it can only reduce approximation error compared to uniform subsampling by a constant, and cannot update its notion of importance based on what points it has already selected.

In this work, we develop a scalable, theoretically-sound Bayesian approximation framework with the same level of automation as ADVI and NUTS, the algorithmic simplicity and low computational burden of Bayesian coresets, and the inferential performance of hand-tuned, model-specific scalable algorithms. We begin with an intuitive reformulation of Bayesian coreset construction as sparse vector sum approximation, in which the data log-likelihood functions are vectors in a vector space, sensitivity is a weighted uniform (i.e. supremum) norm on those vectors, and the construction algorithm is importance sampling. This perspective illuminates the use of the uniform norm as the primary source of the shortcomings of Bayesian coresets. To address these issues we develop Hilbert coresets, i.e., Bayesian coresets using a norm induced by an inner-product on the log-likelihood function space. Our contributions include two candidate norms: one a weighted L^2 norm, and another based on the Fisher information distance (Johnson and Barron, 2004). Given these norms, we provide an importance sampling-based coreset construction algorithm and a more aggressive "direction of improvement"-aware coreset construction based on the Frank–Wolfe algorithm (Frank and Wolfe, 1956; Guélat and Marcotte, 1986; Jaggi, 2013; Lacoste-Julien and Jaggi, 2015; Clarkson, 2010). Our contributions include theoretical guarantees relating the performance of both to coreset size. Since the proposed norms and inner-products cannot in general be computed in closed-form, we automate the construction using a random finite-dimensional projection of the log-likelihood functions inspired by Rahimi and Recht (2007). We test Hilbert coresets empirically on multivariate Gaussian inference, logistic regression, Poisson regression, and von Mises-Fisher mixture modeling with both real and synthetic data; these experiments show that Hilbert coresets provide high quality posterior approximations with a significant reduction in the computational cost of inference compared to standard automated inference algorithms. All proofs are deferred to Appendix A.

2. Background

In the general setting of Bayesian posterior inference, we are given a dataset $(y_n)_{n=1}^N$ of N observations, a likelihood $p(y_n|\theta)$ for each observation given the parameter $\theta \in \Theta$, and a prior density $\pi_0(\theta)$ on Θ. We assume throughout that the data are conditionally i.i.d. given θ; the extension of the present work to the nonidentically distributed case is straightforward. The Bayesian posterior is given by the density

$$\pi(\theta) := \frac{1}{Z} \exp(\mathcal{L}(\theta))\pi_0(\theta), \tag{2.1}$$

where the log-likelihood $\mathcal{L}(\theta)$ and marginal likelihood Z are defined by

$$\mathcal{L}_n(\theta) := \log p(y_n \,|\, \theta), \quad \mathcal{L}(\theta) := \sum_{n=1}^N \mathcal{L}_n(\theta), \quad Z := \int \exp(\mathcal{L}(\theta))\pi_0(\theta)\,\mathrm{d}\theta. \tag{2.2}$$

In almost all cases in practice, an exact closed-form expression of π is not available due to the difficulty of computing Z, forcing the use of approximate Bayesian inference algorithms. While Markov chain Monte Carlo (MCMC) algorithms (Gelman et al., 2013, Chapters 11, 12) are often preferred for their theoretical guarantees asymptotic in running time, they are typically computationally intractable for large N. One way to address this is to construct a small, weighted subset of the original dataset whose log-likelihood approximates that of the full dataset, known as a *Bayesian coreset* (Huggins et al., 2016). This coreset can then be passed to a standard MCMC algorithm. The computational savings from running MCMC on a much smaller dataset can allow a much faster inference procedure while retaining the theoretical guarantees of MCMC. In particular, the aim of the Bayesian coresets framework is to find a set of nonnegative weights $w := (w_n)_{n=1}^N$, a small number of which are nonzero, such that the weighted log-likelihood

$$\mathcal{L}(w,\theta) := \sum_{n=1}^N w_n \mathcal{L}_n(\theta) \quad \text{satisfies} \quad |\mathcal{L}(w,\theta) - \mathcal{L}(\theta)| \leq \epsilon \,|\mathcal{L}(\theta)|\,,\ \forall \theta \in \Theta. \tag{2.3}$$

The algorithm proposed by Huggins et al. (2016) to construct a Bayesian coreset is as follows. First, compute the *sensitivity* σ_n of each data point,

$$\sigma_n := \sup_{\theta\in\Theta} \left| \frac{\mathcal{L}_n(\theta)}{\mathcal{L}(\theta)} \right|, \tag{2.4}$$

and then subsample the dataset by taking M independent draws with probability proportional to σ_n (resulting in a coreset of size $\leq M$) via

$$\sigma := \sum_{n=1}^N \sigma_n \qquad (M_1, \ldots, M_N) \sim \mathsf{Multi}\left(M, \left(\frac{\sigma_n}{\sigma}\right)_{n=1}^N\right) \qquad W_n = \frac{\sigma}{\sigma_n}\frac{M_n}{M}. \tag{2.5}$$

Since $\mathbb{E}\left[W_n\right] = 1$, we have that $\mathbb{E}\left[\mathcal{L}(W,\theta)\right] = \mathcal{L}(\theta)$, and we expect that $\mathcal{L}(W,\theta) \to \mathcal{L}(\theta)$ in some sense as M increases. This is indeed the case; Braverman et al. (2016); Feldman and Langberg (2011) showed that with high probability, the coreset likelihood $\mathcal{L}(W,\theta)$ satisfies Eq. (2.3) with $\epsilon^2 = O\left(\frac{1}{M}\right)$, and Huggins et al. (2016) extended this result to the case of Bayesian coresets in the setting of logistic regression. Typically, exact computation of the sensitivities σ_n is not tractable, so upper bounds are used instead (Huggins et al., 2016).

3. Coresets as sparse vector sum approximation

This section develops an intuitive perspective of Bayesian coresets as sparse vector sum approximation under a uniform norm, and draws on this perspective to uncover the limitations of the framework and avenues for extension. Consider the vector space of functions $g : \Theta \to \mathbb{R}$ with bounded uniform norm weighted by the total log-likelihood $\mathcal{L}(\theta)$,

$$\|g\| := \sup_{\theta\in\Theta} \left| \frac{g(\theta)}{\mathcal{L}(\theta)} \right|. \tag{3.1}$$

In this space, the data log-likelihood functions $\mathcal{L}_n(\theta)$ have vectors $\mathcal{L}_n$ with norm $\sigma_n := \|\mathcal{L}_n\|$ as defined in Eq. (2.4), the total log-likelihood has vector $\mathcal{L} := \sum_{n=1}^N \mathcal{L}_n$, and the coreset guarantee in Eq. (2.3) corresponds to approximation of $\mathcal{L}$ with the vector $\mathcal{L}(w) := \sum_{n=1}^N w_n \mathcal{L}_n$ under the vector norm with error at most ϵ, i.e. $\|\mathcal{L}(w) - \mathcal{L}\| \leq \epsilon$. Given this formulation, we can write the problem of constructing the best coreset of size M as the minimization of approximation error subject to a constraint on the number of nonzero entries in w,

$$\min_{w \in \mathbb{R}^N} \quad \|\mathcal{L}(w) - \mathcal{L}\|^2 \quad \text{s.t.} \quad w \geq 0, \quad \sum_{n=1}^N \mathbb{1}\left[w_n > 0\right] \leq M. \tag{3.2}$$

Eq. (3.2) is a convex optimization with binary constraints, and thus is difficult to solve efficiently in general; we are forced to use approximate methods. The uniform Bayesian coresets framework provides one such approximate method, where $\mathcal{L}/N$ is viewed as the expectation of a uniformly random subsample of $(\mathcal{L}_n)_{n=1}^N$, and importance sampling is used to reduce the expected error of the estimate. Choosing importance probabilities proportional to $\sigma_n = \|\mathcal{L}_n\|$ results in a high-probability bound on approximation error given below in Theorem 3.2. The proof of Theorem 3.2 in Appendix A is much simpler than similar results available in the literature (Feldman and Langberg, 2011; Braverman et al., 2016; Huggins et al., 2016) due to the present vector space formulation. Theorem 3.2 depends on two constants (σ and $\overline{\eta}$) that capture important aspects of the geometry of the optimization problem:

$$\sigma_n := \|\mathcal{L}_n\| \qquad \sigma := \sum_{n=1}^N \sigma_n \qquad \overline{\eta}^2 := \max_{n,m \in [N]} \left\| \frac{\mathcal{L}_n}{\sigma_n} - \frac{\mathcal{L}_m}{\sigma_m} \right\|^2, \tag{3.3}$$

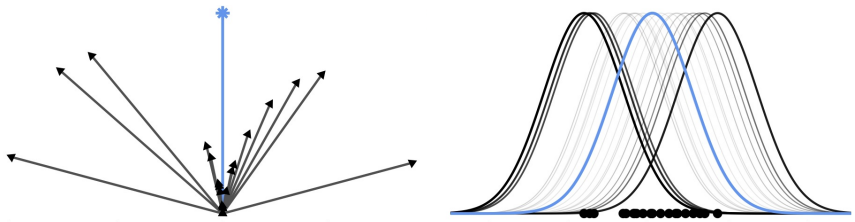
where $[N] := \{1, 2, \ldots, N\}$. The quantity $\sigma \geq 0$ captures the scale of the problem; all error guarantees on $\|\mathcal{L}(w) - \mathcal{L}\|$ should be roughly linearly proportional to σ. The quantity $0 \leq \overline{\eta} \leq 2$ captures how well-aligned the vectors $(\mathcal{L}_n)_{n=1}^N$ are, and thus the inherent difficulty of approximating $\mathcal{L}$ with a sparse weighted subset $\mathcal{L}(w)$. For example, if all vectors are aligned then $\overline{\eta} = 0$, and the problem is trivial since we can achieve 0 error with a single scaled vector $\mathcal{L}_n$. Theorem 3.2 also depends on an approximate notion of the dimension of the span of the log-likelihood vectors $(\mathcal{L}_n)_{n=1}^N$, given by Definition 3.1. Note in particular that the approximate dimension of a set of vectors in $\mathbb{R}^d$ is at most d, corresponding to the usual notion of dimension in this setting.

Definition 3.1. The *approximate dimension* $\dim\left(u_n\right)_{n=1}^N$ of N vectors in a normed vector space is the minimum value of $d \in \mathbb{N}$ such that all vectors u_n can be approximated using linear combinations of a set of d unit vectors $(v_j)_{j=1}^d$, $\|v_j\| = 1$:

$$\forall\, n \in [N],\ \exists\, \alpha \in [-1, 1]^d \text{ s.t. } \left\| \frac{u_n}{\|u_n\|} - \sum_{j=1}^d \alpha_j v_j \right\| \leq \frac{d}{\sqrt{N}}. \tag{3.4}$$

Theorem 3.2. *Fix any $\delta \in (0, 1)$. With probability $\geq 1 - \delta$, the output of the uniform coreset construction algorithm in Eq. (2.5) satisfies*

$$\|\mathcal{L}(W) - \mathcal{L}\| \leq \frac{\sigma}{\sqrt{M}} \left(\dim\left(\mathcal{L}_n\right)_{n=1}^N + \overline{\eta} \sqrt{2 \log \frac{1}{\delta}} \right). \tag{3.5}$$



(A) (B)

FIGURE 1. (1a): The sparse vector approximation problem, depicting the sum $\mathcal{L}$ in blue and the vectors $\mathcal{L}_n$ in grey. (1b): Uniform Bayesian coresets behavior on the simple exercise of learning a Gaussian mean. Depicted are data likelihoods in black, scaled posterior density in blue, and data as black scatter points. Sensitivity is indicated by likelihood opacity and thickness.

The analysis, discussion, and algorithms presented to this point are independent of the particular choice of norm given in Eq. (3.1); one might wonder if the uniform norm used above is the best choice, or if there is another norm more suited to Bayesian inference in some way. For instance, the supremum in Eq. (3.1) can diverge in an unbounded or infinite-dimensional parameter space Θ, requiring an artificial restriction placed on the space (Huggins et al., 2016). This precludes the application to the many common models and priors that have unbounded parameter spaces, even logistic regression with full support $\Theta = \mathbb{R}^d$. The optimization objective function in Eq. (3.1) is also typically nonconvex, and finding (or bounding) the optimum is a model-specific task that is not easily automated.

Perhaps most importantly, the uniform norm lacks a sense of "directionality" as it does not correspond to an inner-product. This implies that the bound in Theorem 3.2 does not scale properly with the alignment of vectors and that it depends on the approximate dimension (which may be hard to compute). Moreover, the lack of directionality makes the coreset construction algorithm behave counterintuitively and limits its performance in a fundamental way. Fig. 1a provides a pictorial representation of this limitation. Recall that the goal of coreset construction is to find a *sparse* weighted subset of the vectors $(\mathcal{L}_n)_{n=1}^N$ (grey) that approximates $\mathcal{L}$ (blue). In this example, there are vectors which, when scaled, could individually nearly perfectly replicate $\mathcal{L}$. But the importance sampling algorithm in Eq. (2.5) will instead tend to sample those vectors with large norm that are pointed away from $\mathcal{L}$, requiring a much larger coreset to achieve the same approximation error. This is a consequence of the lack of directionality of the uniform norm; it has no concept of the alignment of certain vectors with $\mathcal{L}$, and is forced to mitigate worst-case error by sampling those vectors with large norm. Fig. 1b shows the result of this behavior in a 1D Gaussian inference problem. In this figure, the likelihood functions of the data are depicted in black, with their uniform norm (or sensitivity) indicated by thickness and opacity. The posterior distribution is displayed in blue, with its log-density scaled by $1/N$ for clarity. The importance sampling algorithm in Eq. (2.5) will tend to sample those data that are *far away* from the posterior mean, with likelihoods that are different than the scaled posterior, despite the fact that there are data close to the mean whose likelihoods are near-perfect approximations of the scaled

posterior. Using the intuition from Fig. 1, it is not difficult to construct examples where the expected error of importance sampling is arbitrarily worse than the error of the optimal coreset of size M.

4. Hilbert coresets

It is clear that a notion of directionality of the vectors $(\mathcal{L}_n)_{n=1}^N$ is key to developing both efficient, intuitive coreset construction algorithms and theory that correctly reflects problem difficulty. Therefore, in this section we develop methods for constructing Bayesian coresets in a Hilbert space (*Hilbert coresets*), i.e., using a norm corresponding to an inner product. The notion of directionality granted by the inner product provides two major advantages over uniform coresets: coreset points can be chosen intelligently based on the residual posterior approximation error vector; and theoretical guarantees on approximation quality can directly incorporate the difficulty of the approximation problem via the alignment of log-likelihood vectors. We provide two coreset construction algorithms which take advantage of these benefits. The first method, developed in Section 4.1, is based on viewing $\mathcal{L}/N$ as the expectation of a uniformly random subsample of $(\mathcal{L}_n)_{n=1}^N$, and then using importance sampling to reduce the expected error of the estimate. The second method, developed in Section 4.2, is based on viewing the cardinality-unconstrained version of Eq. (3.2) as a quadratic optimization over an appropriately-chosen polytope, and then using the Frank–Wolfe algorithm (Frank and Wolfe, 1956; Guélat and Marcotte, 1986; Jaggi, 2013) to compute a sparse approximation to the optimum. Theoretical guarantees on posterior approximation error are provided for both. In Section 4.3, we develop streaming/distributed extensions of these methods and provide similar approximation guarantees. Note that this section treats the general case of Bayesian coreset construction with a Hilbert space norm; the selection of a particular norm and its automated computation is left to Section 5.

4.1. Coreset construction via importance sampling.

Taking inspiration from the uniform Bayesian coreset construction algorithm, the first Hilbert coreset construction method, Algorithm 1, involves i.i.d. sampling from the vectors $(\mathcal{L}_n)_{n=1}^N$ with probabilities $(p_n)_{n=1}^N$ and reweighting the subsample. In contrast to the case of the weighted uniform norm in Eq. (3.1), the choice $p_n \propto \sigma_n$ exactly minimizes the expected squared coreset error under a Hilbert norm (see Eq. (A.31) in Appendix A), yielding

$$\mathbb{E}\left[\|\mathcal{L}(W) - \mathcal{L}\|^2\right] = \frac{\sigma^2\eta^2}{M} \qquad \eta^2 := 1 - \frac{\|\mathcal{L}\|^2}{\sigma^2}, \tag{4.1}$$

where $0 \leq \eta \leq 1$, similar to $\overline{\eta}$, captures how well-aligned the vectors $(\mathcal{L}_n)_{n=1}^N$ are. However, in a Hilbert space η is a tighter constant: $\eta \leq \overline{\eta}/\sqrt{2}$ by Lemma A.4. Theorem 4.1, whose proof in Appendix A relies on standard martingale concentration inequalities, provides a high-probability guarantee on the quality of the output approximation. This result depends on $\overline{\eta}$ from Eq. (3.3) and η from Eq. (4.1).

Theorem 4.1. *Fix any $\delta \in (0, 1)$. With probability $\geq 1 - \delta$, the output W of Algorithm 1 satisfies*

$$\|\mathcal{L}(W) - \mathcal{L}\| \leq \frac{\sigma}{\sqrt{M}}\left(\eta + \eta_M\sqrt{2\log\frac{1}{\delta}}\right) \tag{4.2}$$

Algorithm 1 IS: Hilbert coresets via importance sampling

Require: $(\mathcal{L}_n)_{n=1}^N$, M, $\|\cdot\|$
$\forall n \in [N]$ $\sigma_n \leftarrow \|\mathcal{L}_n\|$, and $\sigma \leftarrow \sum_{n=1}^N \sigma_n$ ▷ compute norms
$(M_1, \dots, M_N) \leftarrow \mathsf{Multi}\left(M, \left(\frac{\sigma_n}{\sigma}\right)_{n=1}^N\right)$ ▷ subsample the data
$W_n \leftarrow \frac{\sigma}{\sigma_n}\frac{M_n}{M}$ for $n \in [N]$ ▷ reweight the subsample
return W

where

$$\eta_M := \min\left(\overline{\eta}, \eta\sqrt{\frac{2M\eta^2}{\overline{\eta}^2 \log\frac{1}{\delta}}}\, H^{-1}\left(\frac{\overline{\eta}^2 \log\frac{1}{\delta}}{2M\eta^2}\right)\right) \tag{4.3}$$

$$H(y) := (1+y)\log(1+y) - y. \tag{4.4}$$

In contrast to Theorem 3.2, Theorem 4.1 takes advantage of the inner product to incorporate a notion of problem difficulty into the bound. For example, since $\lim_{y\to 0}\sqrt{y^{-1}}H^{-1}(y) = 1$ and $\eta \leq \overline{\eta}$, we have $\lim_{M\to\infty}\eta_M = \eta$, and so the bound in Theorem 4.1 is asymptotically equivalent to $\frac{\sigma\eta}{\sqrt{M}}\left(1 + \sqrt{2\log\frac{1}{\delta}}\right)$ as $M \to \infty$. Given that importance sampling can only improve convergence over uniformly random subsampling by a constant, this constant reduction is significant. Moreover, note that since $H(y) \sim y\log y$ as $y \to \infty$, we have $\lim_{\eta\to 0}\eta_M = 0$, so the bound in Theorem 4.1 approaches 0 as $\eta \to 0$. This behavior correctly captures the fact that the problem gets easier as $\eta \to 0$ and is trivial when $\eta = 0$; this would be the case, e.g., if $\mathcal{L}_n$ are all aligned but have different norms, and Algorithm 1 produces the exact solution for any $M \geq 1$. Note that Theorem 4.1, in conjunction with the fact that $\eta \leq \overline{\eta}$, immediately implies the simpler result in Corollary 4.2.

Corollary 4.2. *Fix any $\delta \in (0,1)$. With probability $\geq 1-\delta$, the output W of Algorithm 1 satisfies*

$$\|\mathcal{L}(W) - \mathcal{L}\| \leq \frac{\sigma\overline{\eta}}{\sqrt{M}}\left(1 + \sqrt{2\log\frac{1}{\delta}}\right). \tag{4.5}$$

4.2. **Coreset construction via Frank–Wolfe.** The major advantages of Algorithm 1 are its simplicity and sole requirement of computing the norms $(\|\mathcal{L}_n\|)_{n=1}^N$. Like the original uniform Bayesian coreset algorithm in Eq. (2.5), however, it does not take into account the residual error in the coreset approximation in order to choose new samples intelligently. The second Hilbert coreset construction method, Algorithm 2, takes advantage of the directionality of the Hilbert norm to incrementally build the coreset by selecting vectors aligned with the "direction of greatest improvement."

The development of Algorithm 2 involves two major steps. First, we replace the cardinality constraint on w in Eq. (3.2) with a polytope constraint:

$$\min_{w\in\mathbb{R}^N} \quad (w-1)^T K(w-1) \quad \text{s.t.} \quad w \geq 0, \quad \sum_{n=1}^N \sigma_n w_n = \sigma, \tag{4.6}$$

where $K \in \mathbb{R}^{N\times N}$ is a kernel matrix defined by $K_{ij} := \langle \mathcal{L}_i, \mathcal{L}_j\rangle$, and we take advantage of the Hilbert norm to rewrite $\|\mathcal{L}(w) - \mathcal{L}\|^2 = (w-1)^T K(w-1)$. The polytope is designed to contain the point $w = 1 := [1, 1, \dots, 1]^T \in \mathbb{R}^N$—which is

Algorithm 2 FW: Hilbert coresets via Frank–Wolfe

Require: $(\mathcal{L}_n)_{n=1}^N$, M, $\langle\cdot,\cdot\rangle$
  $\forall n \in [N]$ $\sigma_n \leftarrow \sqrt{\langle \mathcal{L}_n, \mathcal{L}_n\rangle}$, and $\sigma \leftarrow \sum_{n=1}^N \sigma_n$ ▷ compute norms
  $f \leftarrow \arg\max_{n\in[N]} \left\langle \mathcal{L}, \frac{1}{\sigma_n}\mathcal{L}_n \right\rangle$ ▷ greedy initial vertex f selection
  $w \leftarrow \frac{\sigma}{\sigma_f} 1_f$ ▷ initialize w with full weight on f
  for $t \in \{1, \dots, M-1\}$ **do**
    $f \leftarrow \arg\max_{n\in[N]} \left\langle \mathcal{L} - \mathcal{L}(w), \frac{1}{\sigma_n}\mathcal{L}_n \right\rangle$ ▷ find the FW vertex index f
    $\gamma \leftarrow \frac{\left\langle \frac{\sigma}{\sigma_f}\mathcal{L}_f - \mathcal{L}(w), \mathcal{L} - \mathcal{L}(w) \right\rangle}{\left\langle \frac{\sigma}{\sigma_f}\mathcal{L}_f - \mathcal{L}(w), \frac{\sigma}{\sigma_f}\mathcal{L}_f - \mathcal{L}(w) \right\rangle}$ ▷ closed-form line search for step size γ
    $w \leftarrow (1-\gamma)w + \gamma \frac{\sigma}{\sigma_f} 1_f$ ▷ add/reweight data point f in coreset
  end for
  return w

optimal with cost 0 since $\mathcal{L}(1) = \mathcal{L}$—and have vertices $\frac{\sigma}{\sigma_n}1_n$ for $n \in [N]$, where 1_n is the n^{th} coordinate unit vector. Next, taking inspiration from Clarkson (2010) and Jaggi (2013), we solve the optimization in Eq. (4.6) using the Frank–Wolfe algorithm (Frank and Wolfe, 1956). Since the polytope was chosen such that its vertices each have a single nonzero component, the algorithm adds at most a single data point to the coreset at each iteration; after initialization followed by $M-1$ iterations, this produces a coreset of size $\leq M$.

We initialize w_0 to the vertex most aligned with $\mathcal{L}$, i.e.

$$w_0 = \frac{\sigma}{\sigma_{f_0}} 1_{f_0} \quad \text{where} \quad f_0 = \arg\max_{n\in[N]} \left\langle \mathcal{L}, \frac{1}{\sigma_n}\mathcal{L}_n \right\rangle. \tag{4.7}$$

Let w_t be the iterate at step t. The gradient of the cost is $2K(w_t - 1)$, so the Frank–Wolfe direction is

$$d_t := \frac{\sigma}{\sigma_{f_t}} 1_{f_t} - w_t \quad \text{where} \quad f_t = \arg\max_{n\in[N]} \left\langle \mathcal{L} - \mathcal{L}(w_t), \frac{1}{\sigma_n}\mathcal{L}_n \right\rangle. \tag{4.8}$$

The Frank–Wolfe algorithm applied to Eq. (4.6) thus corresponds to a simple greedy approach in which we select the vector $\mathcal{L}_{f_t}$ with direction most aligned with the residual error $\mathcal{L} - \mathcal{L}(w_t)$. We perform line search to update $w_{t+1} = w_t + \gamma d_t$ for some $\gamma \in [0, 1]$. Since the objective is quadratic, the exact solution for unconstrained line search is available in closed form per Eq. (4.9); Lemma 4.3 shows that this is actually the solution to constrained line search in $\gamma \in [0, 1]$, ensuring that w_{t+1} remains feasible.

$$w_{t+1} = w_t + \gamma_t d_t \quad \text{where} \quad \gamma_t = \frac{\left\langle \frac{\sigma}{\sigma_{f_t}}\mathcal{L}_{f_t} - \mathcal{L}(w_t), \mathcal{L} - \mathcal{L}(w_t) \right\rangle}{\left\langle \frac{\sigma}{\sigma_{f_t}}\mathcal{L}_{f_t} - \mathcal{L}(w_t), \frac{\sigma}{\sigma_{f_t}}\mathcal{L}_{f_t} - \mathcal{L}(w_t) \right\rangle}. \tag{4.9}$$

Lemma 4.3. *For all $t \in \mathbb{N}$, $\gamma_t \in [0, 1]$.*

Theorem 4.4 below provides a guarantee on the quality of the approximation output by Algorithm 2 using the combination of the initialization in Eq. (4.7) and exact line search in Eq. (4.9). This result depends on the constants $\overline{\eta}$ from Eq. (3.3),

η from Eq. (4.1), and ν, defined by

$$\nu^2 := 1 - \frac{r^2}{\sigma^2 \overline{\eta}^2}, \tag{4.10}$$

where r is the distance from $\mathcal{L}$ to the nearest boundary of the convex hull of $\{\sigma\mathcal{L}_n/\sigma_n\}_{n=1}^N$. Since $\mathcal{L}$ is in the relative interior of this convex hull by Lemma A.5, we are guaranteed that $\nu < 1$. The proof of Theorem 4.4 in Appendix A relies on a technique from Guélat and Marcotte (1986, Theorem 2) and a novel bound on the logistic equation.

Theorem 4.4. *The output w of Algorithm 2 satisfies*

$$\|\mathcal{L}(w) - \mathcal{L}\| \leq \frac{\sigma\eta\overline{\eta}\nu}{\sqrt{\overline{\eta}^2\nu^{-2(M-2)} + \eta^2(M-1)}} \leq \frac{\sigma\overline{\eta}}{\sqrt{M}}. \tag{4.11}$$

In contrast to previous convergence analyses of Frank–Wolfe optimization, Theorem 4.4 exploits the quadratic objective and exact line search to capture both the logarithmic $1/\sqrt{M}$ convergence rate for small values of M, and the linear ν^M rate for large M. Alternatively, one can remove the computational cost of computing the exact line search via Eq. (4.9) by simply setting $\gamma_t = \frac{2}{3t+4}$. In this case, Theorem 4.4 is replaced with the weaker result (see the note at the end of the proof of Theorem 4.4 in Appendix A)

$$\|\mathcal{L}(w) - \mathcal{L}\| \leq \frac{2\sigma\overline{\eta}}{\sqrt{3M+1}}. \tag{4.12}$$

4.3. **Distributed coreset construction.** An advantage of Hilbert coresets—and coresets in general—is that they apply to streaming and distributed data with little modification, and retain their theoretical guarantees. In particular, if the dataset $(y_n)_{n=1}^N$ is distributed (not necessarily evenly) among C processors, and either Algorithm 1 or Algorithm 2 are run for M iterations on each processor, the resulting merged coreset of size $\leq MC$ has an error guarantee given by Corollary 4.5 or Corollary 4.6. Note that the weights from each distributed coreset are not modified when merging. These results both follow from Theorems 4.1 and 4.4 with straightforward usage of the triangle inequality, the union bound, and the fact that $\overline{\eta}$ for each subset is bounded above by $\overline{\eta}$ for the full dataset.

Corollary 4.5. *Fix any $\delta \in (0,1)$. With probability $\geq 1-\delta$, the coreset constructed by running Algorithm 1 on C nodes and merging the result satisfies*

$$\|\mathcal{L}(w) - \mathcal{L}\| \leq \frac{\sigma\overline{\eta}}{\sqrt{M}}\left(1 + \sqrt{2\log\frac{C}{\delta}}\right). \tag{4.13}$$

Corollary 4.6. *The coreset constructed by running Algorithm 2 on C nodes and merging the results satisfies*

$$\|\mathcal{L}(w) - \mathcal{L}\| \leq \frac{\sigma\overline{\eta}}{\sqrt{M}}. \tag{4.14}$$

5. Norms and random projection

The algorithms and theory in Section 4 address the scalability and performance of Bayesian coreset construction, but are specified for an arbitrary Hilbert norm; it remains to choose a norm suitable for automated Bayesian posterior approximation. There are two main desiderata for such a norm: it should be a good indicator of

posterior discrepancy, and it should be efficiently computable or approximable in such a way that makes Algorithms 1 and 2 efficient for large N, i.e., $O(N)$ time complexity. To address the desideratum that the norm is an indicator of posterior discrepancy, we propose the use of one of two Hilbert norms. It should, however, be noted that these are simply reasonable suggestions, and other Hilbert norms could certainly be used in the algorithms set out in Section 4. The first candidate is the expectation of the squared 2-norm difference between the log-likelihood gradients under a weighting distribution $\hat{\pi}$,

$$\|\mathcal{L}(w) - \mathcal{L}\|^2_{\hat{\pi},F} := \mathbb{E}_{\hat{\pi}}\left[\|\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}(w,\theta)\|_2^2\right], \tag{5.1}$$

where the weighting distribution $\hat{\pi}$ has the same support as the posterior π. This norm is a weighted version of the Fisher information distance (Johnson and Barron, 2004). The inner product induced by this norm is defined by

$$\langle \mathcal{L}_n, \mathcal{L}_m\rangle_{\hat{\pi},F} := \mathbb{E}_{\hat{\pi}}\left[\nabla\mathcal{L}_n(\theta)^T\nabla\mathcal{L}_m(\theta)\right]. \tag{5.2}$$

Although this norm has a connection to previously known discrepancies between probability distributions, it does require that the likelihoods are differentiable. One could instead employ a simple weighted L^2 norm on the log-likelihoods, given by

$$\|\mathcal{L}(w) - \mathcal{L}\|_{\hat{\pi},2} := \mathbb{E}_{\hat{\pi}}\left[\left(\mathcal{L}(\theta) - \mathcal{L}(w,\theta)\right)^2\right] \tag{5.3}$$

with induced inner product

$$\langle \mathcal{L}_n, \mathcal{L}_m\rangle_{\hat{\pi},2} := \mathbb{E}_{\hat{\pi}}\left[\mathcal{L}_n(\theta)\mathcal{L}_m(\theta)\right]. \tag{5.4}$$

In both cases, the weighting distribution $\hat{\pi}$ would ideally be chosen equal to π to emphasize discrepancies that are in regions of high posterior mass. Though we do not have access to the true posterior without incurring significant computational cost, there are many practical options for setting $\hat{\pi}$, including: the Laplace approximation (Bishop, 2006, Section 4.4), a posterior based on approximate sufficient statistics (Huggins et al., 2017), a discrete distribution based on samples from an MCMC algorithm run on a small random subsample of the data, the prior, independent posterior conditionals (see Eq. (7.6) in Section 7.3), or any other reasonable method for finding a low-cost posterior approximation. This requirement of a low-cost approximation is not unusual, as previous coreset formulations have required similar preprocessing to compute sensitivities, e.g., a k-clustering of the data (Huggins et al., 2016; Lucic et al., 2016; Braverman et al., 2016). We leave the general purpose selection of a weighting function $\hat{\pi}$ for future work.

The two suggested norms $\|\cdot\|_{\hat{\pi},2/F}$ often do not admit exact closed-form evaluation due to the intractable expectations in Eqs. (5.2) and (5.4). Even if closed-form expressions are available, Algorithm 2 is computationally intractable when we only have access to inner products between pairs of individual log-likelihoods $\mathcal{L}_n$, $\mathcal{L}_m$, since obtaining the Frank–Wolfe direction involves the $O(N^2)$ computation $\arg\max_{n\in[N]}\sum_{m=1}^N \langle \mathcal{L}_m, \mathcal{L}_n/\sigma_n\rangle$. Further, the analytic evaluation of expectations is a model- (and $\hat{\pi}$-) specific procedure that cannot be easily automated. To both address these issues and automate Hilbert coreset construction, we use *random features* (Rahimi and Recht, 2007), i.e. a random projection of the vectors $(\mathcal{L}_n)_{n=1}^N$ into a finite-dimensional vector space using samples from $\hat{\pi}$. For the weighted Fisher information inner product in Eq. (5.2), we approximate $\langle \mathcal{L}_n, \mathcal{L}_m\rangle_{\hat{\pi},F}$ with an

Algorithm 3 Bayesian Hilbert coresets with random projection

Require: $(\mathcal{L}_n)_{n=1}^N$, $\hat{\pi}$, M, J
▷ sample feature points and gradient dimension indices
$(\mu_j)_{j=1}^J \overset{\text{i.i.d.}}{\sim} \hat{\pi}$, $(d_j)_{j=1}^J \overset{\text{i.i.d.}}{\sim} \mathsf{Unif}(\{1,\dots,D\})$
▷ construct the random projection using one of the norms from Eqs. (5.7) and (5.10)
If $\|\cdot\|_{\hat{\pi},F}$: $\forall n \in [N]$, $v_n \leftarrow \sqrt{D/J}\,[(\nabla\mathcal{L}_n(\mu_1))_{d_1},\dots,(\nabla\mathcal{L}_n(\mu_J))_{d_J}]^T$
If $\|\cdot\|_{\hat{\pi},2}$: $\forall n \in [N]$, $v_n \leftarrow \sqrt{1/J}\,[\mathcal{L}_n(\mu_1),\dots,\mathcal{L}_n(\mu_J)]^T$
▷ return the coreset constructed using random feature vectors
return $\mathrm{FW}\left((v_n)_{n=1}^N, M, (\cdot)^T(\cdot)\right)$ or $\mathrm{IS}\left((v_n)_{n=1}^N, M, \|\cdot\|_2\right)$

unbiased estimate given by

$$(d_j)_{j=1}^J \overset{\text{i.i.d.}}{\sim} \mathsf{Unif}(\{1,\dots,D\}) \quad (\mu_j)_{j=1}^J \overset{\text{i.i.d.}}{\sim} \hat{\pi} \tag{5.5}$$

$$\langle \mathcal{L}_n, \mathcal{L}_m \rangle_{\hat{\pi},F} \approx \frac{D}{J}\sum_{j=1}^J (\nabla\mathcal{L}_n(\mu_j))_{d_j}(\nabla\mathcal{L}_m(\mu_j))_{d_j}, \tag{5.6}$$

where subscripts indicate the selection of a component of a vector. If we define the J-dimensional vector

$$\hat{\mathcal{L}}_n := \sqrt{\frac{D}{J}}\,[(\nabla\mathcal{L}_n(\mu_1))_{d_1}, (\nabla\mathcal{L}_n(\mu_2))_{d_2},\dots,(\nabla\mathcal{L}_n(\mu_J))_{d_J}]^T, \tag{5.7}$$

we have that for all $n, m \in [N]$,

$$\langle \mathcal{L}_n, \mathcal{L}_m \rangle_{\hat{\pi},F} \approx \hat{\mathcal{L}}_n^T \hat{\mathcal{L}}_m. \tag{5.8}$$

Therefore, for $n \in [N]$, $\hat{\mathcal{L}}_n$ serves as a random finite-dimensional projection of $\mathcal{L}_n$ that can be used in Algorithms 1 and 2. Likewise, for the weighted L^2 inner product in Eq. (5.4), we have

$$(\mu_j)_{j=1}^J \overset{\text{i.i.d.}}{\sim} \hat{\pi} \quad \langle \mathcal{L}_n, \mathcal{L}_m \rangle_{\hat{\pi},2} \approx \frac{1}{J}\sum_{j=1}^J \mathcal{L}_n(\mu_j)\mathcal{L}_m(\mu_j), \tag{5.9}$$

and so defining

$$\hat{\mathcal{L}}_n := \sqrt{\frac{1}{J}}\,[\mathcal{L}_n(\mu_1), \mathcal{L}_n(\mu_2),\dots,\mathcal{L}_n(\mu_J)]^T, \tag{5.10}$$

we have that for all $n, m \in [N]$,

$$\langle \mathcal{L}_n, \mathcal{L}_m \rangle_{\hat{\pi},2} \approx \hat{\mathcal{L}}_n^T \hat{\mathcal{L}}_m, \tag{5.11}$$

and once again $\forall n \in [N]$, $\hat{\mathcal{L}}_n$ serves as a finite-dimensional approximation of $\mathcal{L}_n$. The construction of the random projections is both easily automated and enables the efficient computation of inner products with vector sums. For example, to obtain the Frank–Wolfe direction, rather than computing $\arg\max_{n\in[N]} \sum_{m=1}^N \langle \mathcal{L}_m, \mathcal{L}_n/\sigma_n\rangle$, we can simply compute $\hat{\mathcal{L}} = \sum_{n=1}^N \hat{\mathcal{L}}_n$ in $O(NJ)$ time once at the start of the algorithm and then $\arg\max_{n\in[N]} \frac{1}{\sigma_n}\hat{\mathcal{L}}^T\hat{\mathcal{L}}_n$ in $O(NJ)$ time at each iteration. Further, since $\hat{\mathcal{L}}_n^T\hat{\mathcal{L}}_m$ is an unbiased estimate of $\langle \mathcal{L}_n, \mathcal{L}_m\rangle_{\hat{\pi},2/F}$, we expect the error of the approximation to decrease with the random finite projection dimension J. Theorem 5.2 (below), whose proof may be found in Appendix A, shows that under

reasonable conditions this is indeed the case: the difference between the true output error and random projection output error shrinks as J increases. Note that Theorem 5.2 is quite loose, due to its reliance on a max-norm quadratic form upper bound.

Definition 5.1. (Boucheron et al., 2013, p. 24) A random variable X is *sub-Gaussian with constant* ξ^2 if

$$\forall \lambda \in \mathbb{R}, \quad \mathbb{E}\left[e^{\lambda X}\right] \leq e^{\frac{\lambda^2 \xi^2}{2}}. \tag{5.12}$$

Theorem 5.2. *Let* $\mu \sim \hat{\pi}$, $d \sim \mathsf{Unif}(\{1, \dots, D\})$, *and suppose* $D\nabla\mathcal{L}_n(\mu)_d \nabla\mathcal{L}_m(\mu)_d$ *(given* $\|\cdot\|_{\hat{\pi},F}$*) or* $\mathcal{L}_n(\mu)\mathcal{L}_m(\mu)$ *(given* $\|\cdot\|_{\hat{\pi},2}$*) is sub-Gaussian with constant* ξ^2. *Fix any* $\delta \in (0,1)$. *With probability* $\geq 1-\delta$, *the output of Algorithm 3 satisfies*

$$\|\mathcal{L} - \mathcal{L}(w)\|^2_{\hat{\pi},2/F} \leq \|\hat{\mathcal{L}} - \hat{\mathcal{L}}(w)\|_2^2 + \|w-1\|_1^2 \sqrt{\frac{2\xi^2}{J} \log \frac{2N^2}{\delta}}. \tag{5.13}$$

6. Synthetic evaluation

In this section, we compare Hilbert coresets to uniform coresets and uniformly random subsampling in a synthetic setting where expressions for the exact and coreset posteriors, along with the KL-divergence between them, are available in closed-form. In particular, the methods are used to perform posterior inference for the unknown mean $\mu \sim \mathcal{N}(0, I)$ of a 2-dimensional multivariate normal distribution with known covariance I from a collection of i.i.d. observations $(y_n)_{n=1}^N$:

$$\mu \sim \mathcal{N}(\mu_0, I) \quad (y_n)_{n=1}^N \mid \mu \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu, I). \tag{6.1}$$

Methods. We ran 1000 trials of data generation followed by uniformly random subsampling (`Rand`), uniform coresets (`Unif`), Hilbert importance sampling (`IS`), and Hilbert Frank–Wolfe (`FW`). For the two Hilbert coreset constructions, we used the weighted Fisher information distance in Eq. (5.1). In this simple setting, the exact posterior distribution is a multivariate Gaussian with mean μ_π and covariance Σ_π given by

$$\mu \mid (y_n)_{n=1}^N \sim \mathcal{N}(\mu_\pi, \Sigma_\pi) \qquad \Sigma_\pi = \frac{1}{1+N} I \qquad \mu_\pi = \Sigma_\pi \left(\mu_0 + \sum_{n=1}^N y_n\right). \tag{6.2}$$

For uniform coreset construction, we subsampled the dataset as per Eq. (2.5), where the sensitivity of y_n (see Appendix C for the derivation) is given by

$$\sigma_n = \frac{1}{N}\left(1 + \frac{(y_n - \bar{y})^T (y_n - \bar{y})}{\frac{1}{N}\sum_{m=1}^N y_m^T y_m - \bar{y}^T \bar{y}}\right), \quad \bar{y} := \frac{1}{N}\sum_{n=1}^N y_n. \tag{6.3}$$

This resulted in a multivariate Gaussian uniform coreset posterior approximation with mean $\hat{\mu}_\pi$ and covariance $\hat{\Sigma}_\pi$ given by

$$\hat{\Sigma}_\pi = \frac{1}{1 + \sum_{n=1}^N W_n} I \qquad \hat{\mu}_\pi = \hat{\Sigma}_\pi \left(\mu_0 + \sum_{n=1}^N W_n y_n\right). \tag{6.4}$$

Generating a uniformly random subsample posterior approximation involved a similar technique, instead using probabilities $\frac{1}{N}$ for all $n \in [N]$. For the Hilbert coreset algorithms, we used the true posterior as the weighting distribution, i.e., $\hat{\pi} = \pi$. This was chosen to illustrate the ideal case in which the true posterior π is

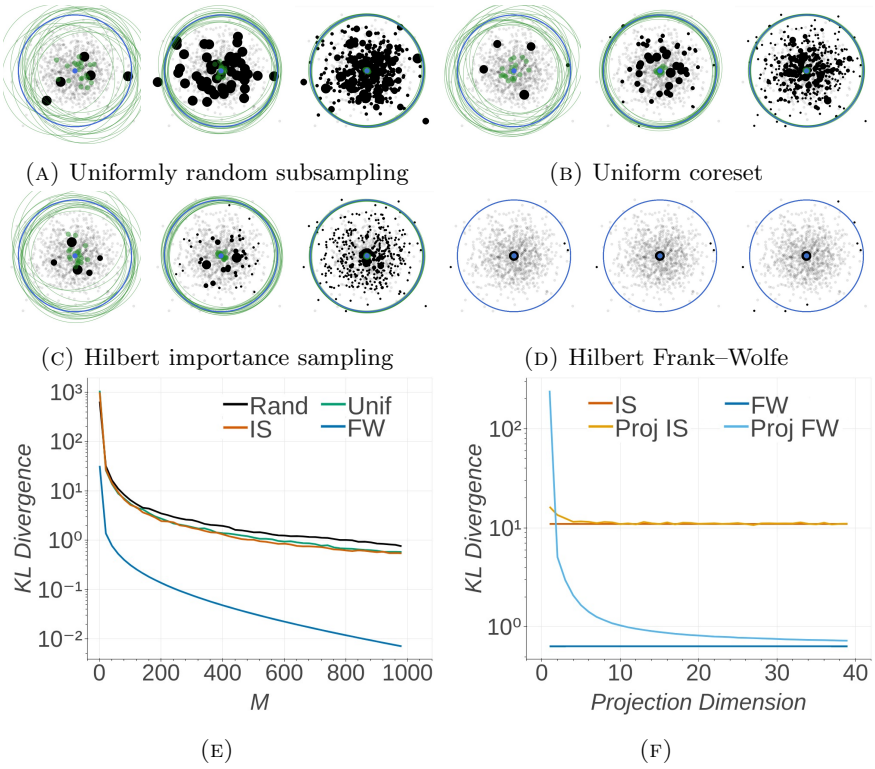


FIGURE 2. (2a-2d): Comparison of different coreset constructions for Gaussian inference, showing example coreset posterior predictive 3σ ellipses (green), the true data generating distribution 3σ ellipse (blue), and a single trace of coreset construction (black) for $M = 5$, 50, and 500. The radius of each coreset point indicates its weight. (2e): A comparison of approximate posteriors using exact norms versus coreset construction iterations M. (2f): A comparison of exact and projected methods versus projection dimension J, with fixed $M = 50$.

well-approximated by the weighting distribution $\hat{\pi}$. Given this choice, the Fisher information distance inner product is available in closed-form:

$$\langle \mathcal{L}_n, \mathcal{L}_m \rangle_{\pi,F} = \frac{2}{1+N} + (\mu_\pi - y_n)^T (\mu_\pi - y_m) . \tag{6.5}$$

Note that the norm $\|\mathcal{L}_n\|_{\pi,F}$ implied by Eq. (6.5) and the uniform sensitivity from Eq. (6.3) are functionally very similar; both scale with the squared distance from y_n to an estimate of μ. Since all the approximate posteriors are multivariate Gaussians of the form Eq. (6.4)—with weights W_n differing depending on the construction algorithm—we were able to evaluate posterior approximation quality exactly using the KL-divergence from the approximate coreset posterior $\tilde{\pi}$ to π, given by

$$\mathrm{D_{KL}}\left(\pi || \tilde{\pi}\right) = \frac{1}{2}\left\{ \mathrm{tr}\left(\Sigma_{\tilde{\pi}}^{-1}\Sigma_\pi\right) + (\mu_{\tilde{\pi}} - \mu_\pi)^T \Sigma_{\tilde{\pi}}^{-1} (\mu_{\tilde{\pi}} - \mu_\pi) - 2 + \log \frac{|\Sigma_{\tilde{\pi}}|}{|\Sigma_\pi|} \right\} . \tag{6.6}$$

Results. The results of this test appear in Fig. 2. The visual comparison of the different coreset constructions in Fig. 2a–2d makes the advantages of Hilbert coresets constructed via Frank–Wolfe clear. As more coreset points are added, all approximate posteriors converge to the true posterior; however, the Frank–Wolfe method requires many fewer coreset points to converge on a reliable estimate. While both the Hilbert and uniform coresets subsample the data favoring those points at greater distance from the center, the Frank–Wolfe method first selects a point close to the center (whose scaled likelihood well-approximates the true posterior), and then refines its estimate with points far away from the center. This intuitive behavior results in a higher-quality approximation of the posterior across all coreset sizes. Note that the black coreset points across $M = 5$, 50, and 500 show a single trace of a coreset being constructed, while the green posterior predictive ellipses show the noise in the coreset construction across multiple runs at each fixed value of M. The quantitative results in Figs. 2e and 2f—which plot the KL-divergence between each coreset approximate posterior and the truth as the projection dimension J or the number of coreset iterations M varies—confirm the qualitative evaluations. In addition, Fig. 2e confirms the theoretical result from Theorem 4.4, i.e., that Frank–Wolfe exhibits linear convergence in this setting. Fig. 2f similarly confirms Theorem 5.2, i.e., that the posterior error of the projected Hilbert coreset converges to that of the exact Hilbert coreset as the dimension J of the random projection increases.

7. Experiments

In this section we evaluate the performance of Hilbert coresets compared with uniform coresets and uniformly random subsampling, using MCMC on the full dataset as a benchmark. We test the algorithms on logistic regression, Poisson regression, and directional clustering models applied to numerous real and synthetic datasets. Based on the results of the synthetic comparison presented in Section 6, for clarity we focus the tests on comparing uniformly random subsampling to Hilbert coresets constructed using Frank–Wolfe with the weighted Fisher information distance from Eq. (5.1). Additional results on importance sampling, uniform coresets, and the weighted 2-norm from Eq. (5.3) are deferred to Appendix B.

7.1. Models.

In the **logistic regression** setting, we are given a set of data points $(x_n, y_n)_{n=1}^N$ each consisting of a feature $x_n \in \mathbb{R}^D$ and a label $y_n \in \{-1, 1\}$, and the goal is to predict the label of a new point given its feature. We thus seek to infer the posterior distribution of the parameter $\theta \in \mathbb{R}^D$ governing the generation of y_n given x_n via

$$\theta \sim \mathcal{N}(0, I) \qquad y_n \,|\, x_n, \theta \overset{\text{indep}}{\sim} \mathsf{Bern}\left(\frac{1}{1 + e^{-z_n^T \theta}}\right) \qquad z_n := [y_n x_n,\, 1]^T. \tag{7.1}$$

In the **Poisson regression** setting, we are given a set of data points $(x_n, y_n)_{n=1}^N$ each consisting of a feature $x_n \in \mathbb{R}^D$ and a count $y_n \in \mathbb{N}$, and the goal is to learn a relationship between features x_n and the associated mean count. We thus seek to infer the posterior distribution of the parameter $\theta \in \mathbb{R}^D$ governing the generation of y_n given x_n via

$$\theta \sim \mathcal{N}(0, I) \qquad y_n \,|\, x_n, \theta \overset{\text{indep}}{\sim} \mathsf{Poiss}\left(\log\left(1 + e^{\theta^T z_n}\right)\right) \qquad z_n := [x_n, 1]^T. \tag{7.2}$$

Finally, in the **directional clustering** setting, we are given a dataset of points $(x_n)_{n=1}^N$ on the unit $(D-1)$-sphere, i.e. $x_n \in \mathbb{R}^D$ with $\|x_n\|_2 = 1$, and the goal is to separate them into K clusters. For this purpose we employ a *von Mises-Fisher (vMF) mixture model* (Banerjee et al., 2005). The component likelihood in this model is the von Mises-Fisher distribution $\text{vMF}(\mu, \tau)$ with concentration $\tau \in \mathbb{R}_+$ and mode $\mu \in \mathbb{R}^D$, $\|\mu\|_2 = 1$, having density

$$f_{\text{vMF}}(x; \mu, \tau) = C_D(\tau) e^{\tau x^T \mu} \qquad C_D(\tau) = \frac{\tau^{D/2-1}}{(2\pi)^{D/2} I_{D/2-1}(\tau)} \tag{7.3}$$

with support on the unit $(D-1)$-sphere $\mathbb{S}^{D-1}$, where I_p denotes the modified Bessel function of the first kind of order p. We place uniform priors on both the component modes and mixture weights, and set $\tau = 50$, resulting in the generative model

$$(\mu_k)_{k=1}^K \overset{\text{i.i.d.}}{\sim} \mathsf{Unif}\left(\mathbb{S}^{D-1}\right) \quad (\omega_k)_{k=1}^K \sim \mathsf{Dir}(1, \ldots, 1) \tag{7.4}$$

$$(x_n)_{n=1}^N \,|\, (\omega_k, \mu_k)_{k=1}^K \overset{\text{i.i.d.}}{\sim} \sum_{k=1}^K \omega_k \text{vMF}(\mu_k, \tau). \tag{7.5}$$

7.2. **Datasets.** We tested the coreset construction methods for each model on a number of datasets. For **logistic regression**, the `Synthetic` dataset consisted of $N = 10{,}000$ data points (with 1,000 held out for testing) with covariate $x_n \in \mathbb{R}^2$ sampled i.i.d. from $\mathcal{N}(0, I)$, and label $y_n \in \{-1, 1\}$ generated from the logistic likelihood with parameter $\theta = [3, 3, 0]^T$. The `Phishing`[2] dataset consisted of $N = 11{,}055$ data points (with 1,105 held out for testing) each with $D = 68$ features. In this dataset, each covariate corresponds to the features of a website, and the goal is to predict whether or not a website is a phishing site. The `ChemReact`[3] dataset consisted of $N = 26{,}733$ data points (with 2,673 held out for testing) each with $D = 10$ features. In this dataset, each covariate represents the features of a chemical experiment, and the label represents whether a chemical was reactive in that experiment or not.

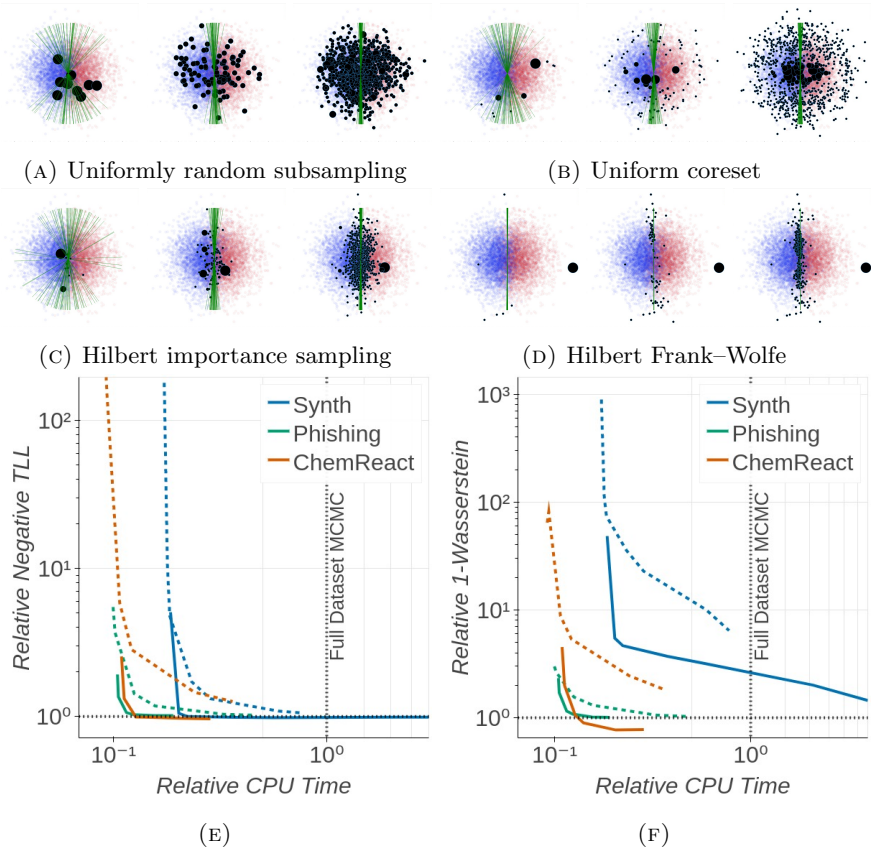
For **Poisson regression**, the `Synthetic` dataset consisted of $N = 10{,}000$ data points (with 1,000 held out for testing) with covariate $x_n \in \mathbb{R}$ sampled i.i.d. from $\mathcal{N}(0, 1)$, and count $y_n \in \mathbb{N}$ generated from the Poisson likelihood with $\theta = [1, 0]^T$. The `BikeTrips`[4] dataset consisted of $N = 17{,}386$ data points (with 1,738 held out for testing) each with $D = 8$ features. In this dataset, each covariate corresponds to weather and season information for a particular hour during the time between 2011–2012, and the count is the number of bike trips taken during that hour in a bikeshare system in Washington, DC. The `AirportDelays`[5] dataset consisted of $N = 7{,}580$ data points (with 758 held out for testing) each with $D = 15$ features. In this dataset, each covariate corresponds to the weather information of a day during the time between 1987–2008, and the count is the number of flights leaving Boston Logan airport delayed by more than 15 minutes that day.

[2] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

[3] http://komarix.org/ac/ds/

[4] http://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset

[5] Airport information from http://stat-computing.org/dataexpo/2009/the-data.html, with historical weather information from https://www.wunderground.com/history/.



(A) Uniformly random subsampling (B) Uniform coreset

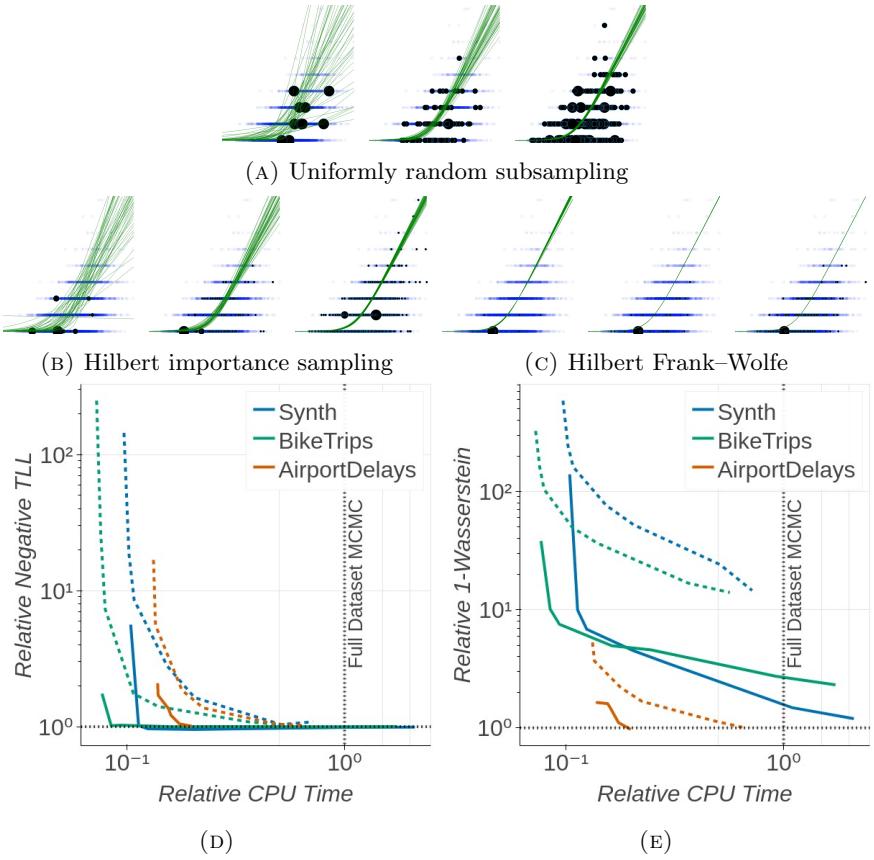
(C) Hilbert importance sampling (D) Hilbert Frank–Wolfe

(E) (F)

FIGURE 3. (3a-3d): Comparison of different coreset constructions for logistic regression on the `Synthetic` dataset (with blue & red labeled data), showing example coreset posterior mean classification boundaries (green), and a single trace of coreset construction (black) for $M = 10$, 100, and 1000. The radius of each coreset point indicates its weight. (3e, 3f): A comparison of negative test log-likelihood (3e) and 1-Wasserstein distance (3f) versus computation time for Frank–Wolfe (solid) and uniform random subsampling (dashed) on the logistic regression model. Both axes are normalized using results from running MCMC on the full dataset; see Section 7.3.

Finally, for **directional clustering**, the `Synthetic` dataset consisted of $N =$ 10,000 data points (with 1,000 held out for testing) generated from an equally-weighted vMF mixture with 6 components, one centered at each of the axis poles.

7.3. **Methods.** We ran 50 trials of uniformly random subsampling and Hilbert Frank–Wolfe using the approximate Fisher information distance in Eq. (5.1), varying $M \in \{10, 50, 100, 500, 1{,}000, 5{,}000, 10{,}000\}$. For both **logistic regression** and **Poisson regression**, we used the Laplace approximation (Bishop, 2006, Section 4.4) as the weighting distribution $\hat{\pi}$ in the Hilbert coreset, with the random projection



(A) Uniformly random subsampling

(B) Hilbert importance sampling

(C) Hilbert Frank–Wolfe

(D)

(E)

FIGURE 4. (4a-4c): Comparison of different coreset constructions for Poisson regression on the `Synthetic` dataset, showing example coreset posterior Poisson mean curves (green), and a single trace of coreset construction (black) for $M = 10$, 100, and 1,000. The radius of each coreset point indicates its weight. (4d, 4e): A comparison of negative test log-likelihood (4d) and 1-Wasserstein distance (4e) versus computation time for Frank–Wolfe (solid) and uniform random subsampling (dashed) on the Poisson regression model. Both axes are normalized using results from running MCMC on the full dataset; see Section 7.3.

dimension set to $D = 500$. Posterior inference in each of the 50 trials was conducted using random walk Metropolis-Hastings with an isotropic multivariate Gaussian proposal distribution. We simulated a total of 100,000 steps, with 50,000 warmup steps including proposal covariance adaptation with a target acceptance rate of 0.234, and thinning of the latter 50,000 by a factor of 5, yielding 10,000 posterior samples.

For **directional clustering** the weighting distribution $\hat{\pi}$ for the Hilbert coreset was constructed by finding maximum likelihood estimates of the cluster modes $(\hat{\mu}_k)_{k=1}^K$ and weights $\hat{\omega}$ using the EM algorithm, and then setting $\hat{\pi}$ to an independent

product of approximate posterior conditionals,

$$\begin{aligned} \bar{x}_k &:= \sum_{n=1}^{N} z_{nk} x_n & \bar{z}_k &:= \sum_{n=1}^{N} z_{nk} \\ \mu_k &\overset{\text{indep}}{\sim} \text{vMF}\left(\frac{\bar{x}_k}{\|\bar{x}_k\|}, \tau\|\bar{x}_k\|\right) & \omega &\overset{\text{indep}}{\sim} \mathsf{Dir}\left(1+\bar{z}_1, \ldots, 1+\bar{z}_k\right), \end{aligned} \tag{7.6}$$

where $(z_n)_{n=1}^N$, $z_n \in \mathbb{R}_+^K$ are the smoothed cluster assignments. The random projection dimension was set to $D = 500$, and the number of clusters K was set to 6. Posterior inference in each of the 50 trials was conducted used Gibbs sampling (introducing auxiliary label variables for the data) with a total of 100,000 steps, with 50,000 warmup steps and thinning of the latter 50,000 by a factor of 5, yielding 10,000 posterior samples. Note that this approach is exact for the full dataset; for the coreset constructions with weighted data, we replicate each data point by its ceiled weight, and then rescale the assignment variables to account for the fractional weight. In particular, for coreset weights $(w_n)_{n=1}^N$, we sample labels for points with $w_n > 0$ via

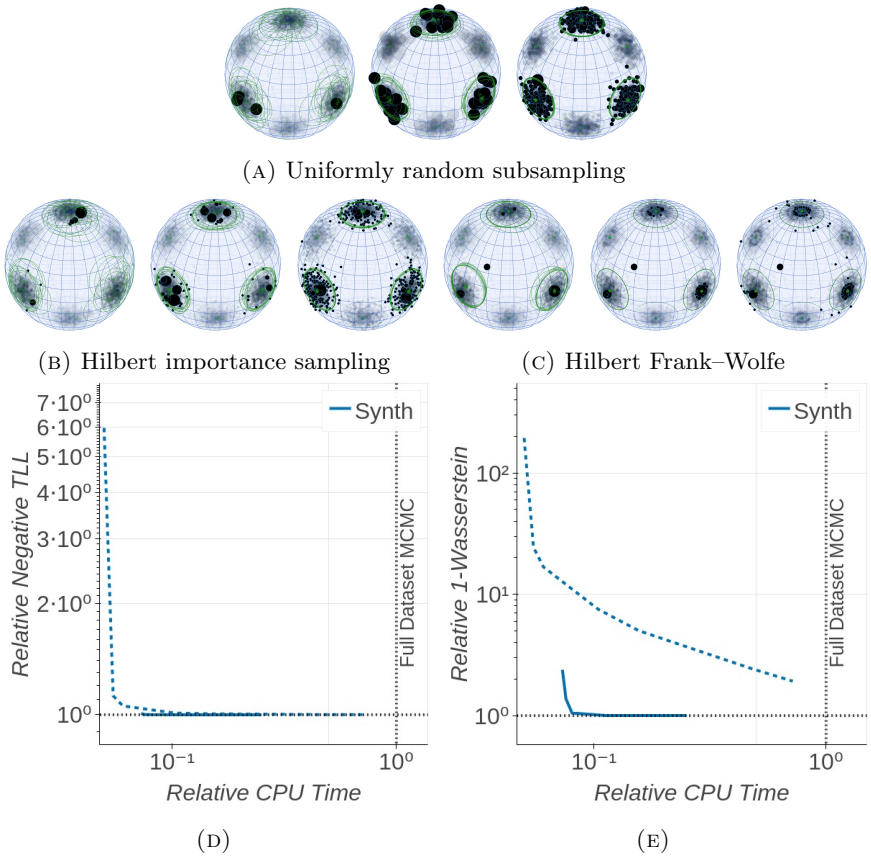
$$\gamma_k \propto \omega_k f_{\text{vMF}}\left(x_n; \mu_k, \tau\right) \quad z_n \overset{\text{indep}}{\sim} \mathsf{Multi}(\texttt{ceil}(w_n), \gamma) \quad z_n \leftarrow \frac{w_n}{\texttt{ceil}(w_n)} z_n, \tag{7.7}$$

and sample the cluster centers and weights via Eq. (7.6).

For all models, we evaluate two metrics of posterior quality: negative log-likelihood on the held-out test set, averaged over posterior MCMC samples; and 1-Wasserstein distance of the posterior samples to samples obtained from running MCMC on the full dataset. All negative test log-likelihood results are shifted by the maximum possible test log-likelihood and normalized by the test log-likelihood obtained from the full dataset posterior. All 1-Wasserstein distance results are normalized by the median pairwise 1-Wasserstein distance between 10 trials of MCMC on the full dataset. All computation times are normalized by the median computation time for MCMC on the full dataset across the 10 trials. These normalizations allow the results from multiple datasets to be plotted coherently on the same axes.

We ran the same experiments described above on Hilbert importance sampling for all datasets, and uniform coresets on the logistic regression model with $a = 3$ and $K = 4$ (see Huggins et al. (2016, Sec. 4.2)). We also compared Hilbert coresets with the weighted 2-norm in Eq. (5.3) to the weighted Fisher information distance in Eq. (5.1). The results of these experiments are deferred to Appendix B for clarity.

7.4. **Results and discussion.** Figs. 3, 4 and 5 show the experimental results for logistic regression, Poisson regression, and directional clustering, respectively. The visual comparisons of coreset construction for all models mimic the results of the synthetic evaluation in Fig. 2. For all the algorithms, the approximate posterior converges to the true posterior on the full dataset as more coreset points are added; and the Frank–Wolfe Hilbert coreset construction selects the most useful points incrementally, creating intuitive coresets that outperform all the other methods. For example, in the logistic regression model, the uniform coreset construction sensitivities are based on the proximity of the data to the centers of a K-clustering, and do not directly incorporate information about the classification boundary. This construction therefore generally favors sampling points on the periphery of the dataset and assigns high weight to those near the center. The Hilbert importance sampling algorithm, in contrast, directly considers the logistic regression problem; it



(A) Uniformly random subsampling

(B) Hilbert importance sampling

(C) Hilbert Frank–Wolfe

(D)

(E)

FIGURE 5. (5a-5c): Comparison of different coreset constructions for directional clustering, showing example coreset posterior mean clusters (green), and a single trace of coreset construction (black) for $M = 10$, 100, and 1,000. The radius of each coreset point indicates its weight. (5d, 5e): A comparison of negative test log-likelihood (5d) and 1-Wasserstein distance (5e) versus computation time for Frank–Wolfe (solid) and uniform random subsampling (dashed) on the directional clustering model. Both axes are normalized using results from running MCMC on the full dataset; see Section 7.3.

favors sampling points lying along the boundary and assigns high weight to points orthogonal to it, thereby fixing the boundary plane more accurately. The Hilbert Frank–Wolfe algorithm selects a single point closely aligned with the classification boundary normal, and then refines its estimate with points near the boundary. This enables it to use far fewer coreset points to achieve a more accurate posterior estimate than the sampling-based methods. Similar statements hold for the two other models: in the Poisson regression model, the Hilbert Frank–Wolfe algorithm chooses a point closest to the true parameter and then refines its estimate using far away points; and in the directional clustering model, the Hilbert Frank–Wolfe algorithm initially selects points near the cluster centers and then refines the estimates with points in

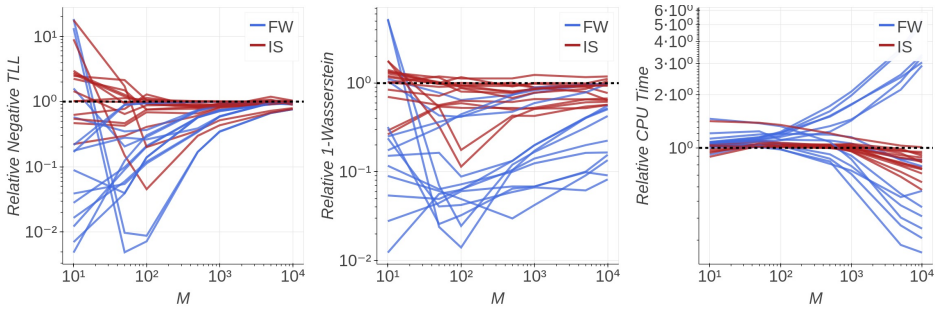


FIGURE 6. A comparison of (6a) negative test log-likelihood, (6b) 1-Wasserstein posterior distance estimate, and (6c) computation time versus coreset construction iterations M for Frank–Wolfe (blue), importance sampling (red), and uniformly random subsampling (dashed black) across all models and datasets. All metrics are normalized to the median value for uniformly random subsampling; see Section 7.3.

each cluster far from its center. The quantitative results demonstrate the strength of the Hilbert Frank–Wolfe coreset construction algorithm: for a given computational time budget, this algorithm provides orders of magnitude reduction in both error metrics over uniformly random subsampling. In addition, the Hilbert Frank–Wolfe coreset achieves the same negative test log-likelihood as the full dataset in roughly a tenth of the computation time. These statements hold across all models and datasets considered.

Fig. 6 provides a summary of the performance of Hilbert coresets as a function of construction iterations M across all models and datasets. This demonstrates its power not only as a scalable inference method but also as a dataset compression technique. For any value of M and across a wide variety of models and datasets, Hilbert coresets (both importance sampling and Frank–Wolfe-based constructions) provide a significant improvement in posterior approximation quality over uniformly random subsampling, with comparable computation time. This figure also shows a rather surprising result: not only does the Frank–Wolfe-based method provide improved posterior estimates, it also can sometimes have *reduced overall computational cost* compared to uniformly random subsampling for fixed M. This is due to the fact that M is an upper bound on the coreset size; the Frank–Wolfe algorithm often selects the same point multiple times, leading to coresets of size $\ll M$, whereas subsampling techniques always have coresets of size $\approx M$. Since the cost of posterior inference scales with the coreset size and dominates the cost of setting up either coreset construction algorithm, the Hilbert Frank–Wolfe method has a reduced overall cost. Generally speaking, we expect the Hilbert coreset methods to be slower than random subsampling for small M, where $\hat{\pi}$ setup and random projection dominates the time cost, but the Frank–Wolfe method to sometimes be faster for large M where the smaller coreset provides a significant inferential boost.

Although more detailed results for Hilbert importance sampling, uniform coresets, and the weighted 2-norm from Eq. (5.3) are deferred to Appendix B, we do provide

a brief summary here. Fig. 7 shows that Hilbert importance sampling provides comparable performance to both uniform coresets and uniformly random subsampling on all models and datasets. The Hilbert Frank–Wolfe coreset construction algorithm typically outperforms all random subsampling methods. Finally, Figs. 8 and 9 show that the weighted 2-norm and Fisher information norm perform similarly in all cases considered.

8. Conclusion

This paper presented a fully-automated, scalable, and theoretically-sound Bayesian inference framework based on Hilbert coresets. The algorithms proposed in this work are simple to implement, reliably provide high-quality posterior approximation at a fraction of the cost of running inference on the full dataset, and enable experts and nonexperts alike to conduct sophisticated modeling and exploratory analysis at scale. There are many avenues for future work, including exploring the application of Hilbert coresets in more complex high-dimensional models, using alternate Hilbert norms, connecting the norms proposed in the present work to more well-known measures of posterior discrepancy, investigating different choices of weighting function, obtaining tighter bounds on the quality of the random projection result, and using variants of the Frank–Wolfe algorithm (e.g. away-step, pairwise, and fully-corrective FW (Lacoste-Julien and Jaggi, 2015)) with stronger convergence guarantees.

Acknowledgments. This research is supported by a Google Faculty Research Award, an MIT Lincoln Laboratory Advanced Concepts Committee Award, and ONR grant N00014-17-1-2072.

Appendix A. Technical results and proofs

In this section we provide proofs of the main results from the paper, along with supporting technical lemmas. Lemma A.1 is the martingale extension of Hoeffding's inequality (Boucheron et al., 2013, Theorem 2.8, p. 34) known as Azuma's inequality. Lemma A.2 is the martingale extension of Bennet's inequality (Boucheron et al., 2013, Theorem 2.9, p. 35). Lemma A.3 provides bounds on the expectation and martingale differences of the norm of a vector constructed by i.i.d. sampling from a discrete distribution. Finally, Lemma A.5 is a geometric result and Lemma A.6 bounds iterates of the logistic equation, both of which are used in the proof of the Frank-Wolfe error bound. Lemma A.4 provides a relationship between two vector alignment constants in the main text.

Lemma A.1 (Azuma's Inequality)**.** *Suppose* $(Y_m)_{m=0}^M$ *is a martingale adapted to the filtration* $(\mathcal{F}_m)_{m=0}^M$*. If there is a constant* ξ *such that for each* $m \in \{1, \dots, M\}$*,*

$$|Y_m - Y_{m-1}| \leq \xi \quad a.s., \tag{A.1}$$

then for all $\epsilon \geq 0$*,*

$$\mathbb{P}\left(Y_M - Y_0 > \epsilon\right) \leq e^{-\frac{\epsilon^2}{2M\xi^2}}. \tag{A.2}$$

Lemma A.2 (Martingale Bennet Inequality)**.** *Suppose* $(Y_m)_{m=0}^M$ *is a martingale adapted to the filtration* $(\mathcal{F}_m)_{m=0}^M$*. If there are constants* ξ *and* τ^2 *such that for each*

$m \in \{1, \dots, M\}$,

$$|Y_m - Y_{m-1}| \leq \xi \qquad \text{and} \qquad \mathbb{E}\left[(Y_m - Y_{m-1})^2 \mid \mathcal{F}_{m-1}\right] \leq \tau^2 \quad \text{a.s.}, \tag{A.3}$$

then for all $\epsilon \geq 0$,

$$\mathbb{P}\left(Y_M - Y_0 > \epsilon\right) \leq e^{-\frac{M\tau^2}{\xi^2} H\left(\frac{\epsilon\xi}{M\tau^2}\right)}, \quad H(x) := (1+x)\log(1+x) - x. \tag{A.4}$$

Lemma A.3. *Suppose* U *and* $\{U_m\}_{m=1}^M$ *are i.i.d. random vectors in a normed vector space with discrete support on* $(u_n)_{n=1}^N$ *with probabilities* $(p_n)_{n=1}^N$*, and*

$$Y := \left\| \frac{1}{M} \sum_{m=1}^{M} U_m - \mathbb{E}\left[U\right] \right\|. \tag{A.5}$$

Then we have the following results.

(1) *If* $\dim (u_n)_{n=1}^N \leq d$ *where* dim *is given by Definition 3.1,*

$$\mathbb{E}\left[Y\right] \leq \frac{d}{\sqrt{M}} \left(\sum_{n=1}^{N} \|u_n\| \sqrt{\frac{p_n(1-p_n)}{N}} + \sqrt{\mathrm{Var}\left[\|U\|\right]} \right). \tag{A.6}$$

(2) *If the norm is a Hilbert norm,*

$$\mathbb{E}\left[Y\right] \leq \frac{1}{\sqrt{M}} \sqrt{\mathbb{E}\left[\|U\|^2\right] - \|\mathbb{E}\left[U\right]\|^2}. \tag{A.7}$$

(3) *The random variable* $Y_m := \mathbb{E}\left[Y \mid \mathcal{F}_m\right]$ *with* $\mathcal{F}_m$ *the* σ*-algebra generated by* $U_1, \dots, U_m$ *is a martingale that satisfies, for* $m \geq 1$*, both*

$$|Y_m - Y_{m-1}| \leq \frac{1}{M} \max_{n,\ell} \|u_n - u_\ell\| \tag{A.8}$$

and

$$\mathbb{E}\left[(Y_m - Y_{m-1})^2 \mid \mathcal{F}_{m-1}\right] \leq \frac{1}{M^2} \mathbb{E}\left[\|U - U_1\|^2\right] \tag{A.9}$$

almost surely.

Proof. (1) Denote the coefficients used to approximate the vector u_n in Definition 3.1 as $\alpha_{n\cdot} \in [-1, 1]^d$. Then using the triangle inequality, denoting the number of times vector u_n is sampled as M_n,

$$\begin{aligned} \mathbb{E}\left[Y\right] \leq & \frac{1}{M} \sum_{n=1}^{N} \frac{d\,\|u_n\|}{\sqrt{N}} \mathbb{E}\left[|M_n - Mp_n|\right] \\ & + \frac{1}{M} \sum_{j=1}^{d} \mathbb{E}\left[\left| \sum_{n=1}^{N} \left(M_n - Mp_n\right) \|u_n\| \alpha_{nj} \right| \right]. \end{aligned} \tag{A.10}$$

Next we bound $\mathbb{E}\left[|\cdot|\right] \leq \sqrt{\mathbb{E}\left[(\cdot)^2\right]}$ via Jensen's inequality, and evaluate the multinomial variances. Defining A_j to be the random variable equal to α_{nj} with probability p_n independently across j, we have that

$$\mathbb{E}\left[Y\right] \leq \frac{1}{\sqrt{M}} \sum_{n=1}^{N} d\,\|u_n\| \sqrt{\frac{p_n(1-p_n)}{N}} + \frac{1}{\sqrt{M}} \sum_{j=1}^{d} \sqrt{\mathrm{Var}\left[\|UA_j\|\right]}. \tag{A.11}$$

The fact that $|A_j| \leq 1$ a.s. yields the desired result.

(2) This follows from Jensen's inequality to write $\mathbb{E}[Y] \leq \sqrt{\mathbb{E}[Y^2]}$ and the expansion of the squared norm.
(3) $(Y_m)_{m=0}^M$ is a standard Doob martingale with $Y_0 = \mathbb{E}[Y]$. Letting $U'_\ell = U_\ell$ for $\ell \neq m$ and U'_m be an independent random variable with $U'_m \overset{d}{=} U_m$, by the triangle inequality we have

$$
\begin{aligned}
&|Y_m - Y_{m-1}| && \text{(A.12)}\\
=&|\mathbb{E}[Y \mid \mathcal{F}_m] - Y_{m-1}| && \text{(A.13)}\\
=&\left|\mathbb{E}\left[\left\|\frac{1}{M}\sum_{\ell=1}^M U_\ell - \mathbb{E}[U]\right\| \mid \mathcal{F}_m\right] - Y_{m-1}\right| && \text{(A.14)}\\
=&\left|\mathbb{E}\left[\left\|\frac{1}{M}(U_m - U'_m) + \frac{1}{M}\sum_{\ell=1}^M U'_\ell - \mathbb{E}[U]\right\| \mid \mathcal{F}_m\right] - Y_{m-1}\right| && \text{(A.15)}\\
\leq&\left|\frac{1}{M}\mathbb{E}[\|U_m - U'_m\| \mid \mathcal{F}_m] + \mathbb{E}\left[\left\|\frac{1}{M}\sum_{\ell=1}^M U'_\ell - \mathbb{E}[U]\right\| \mid \mathcal{F}_m\right] - Y_{m-1}\right| && \text{(A.16)}\\
=&\frac{1}{M}\mathbb{E}[\|U_m - U'_m\| \mid \mathcal{F}_m] && \text{(A.17)}\\
\leq&\frac{1}{M}\max_{n,\ell}\|u_n - u_\ell\|. && \text{(A.18)}
\end{aligned}
$$

Next, using Eq. (A.17) and Jensen's inequality, we have that

$$
\begin{aligned}
\mathbb{E}\left[(Y_m - Y_{m-1})^2 \mid \mathcal{F}_{m-1}\right] &\leq \mathbb{E}\left[\mathbb{E}\left[\frac{1}{M}\|U_m - U'_m\| \mid \mathcal{F}_m\right]^2 \mid \mathcal{F}_{m-1}\right] && \text{(A.19)}\\
&\leq \frac{1}{M^2}\mathbb{E}\left[\|U_m - U'_m\|^2 \mid \mathcal{F}_{m-1}\right] && \text{(A.20)}\\
&= \frac{1}{M^2}\mathbb{E}\left[\|U_m - U'_m\|^2\right]. && \text{(A.21)}
\end{aligned}
$$

□

Proof of Theorem 3.2. Set $\delta \in (0,1)$. Rearranging the results of Lemma A.1, we have that with probability $\geq 1-\delta$,

$$
Y_M \leq Y_0 + \sqrt{2M\xi^2 \log\frac{1}{\delta}}. \tag{A.22}
$$

We now apply the results of Lemma A.3(1) and Lemma A.3(3), noting that $\sum_{n=1}^N \sqrt{p_n(1-p_n)}$ is maximized when $p_n = 1/N$, where the discrete distribution is specified by atoms $u_n = \sigma\mathcal{L}_n/\sigma_n$ with probabilities σ_n/σ for $n \in [N]$:

$$
Y_0 \leq \frac{\sigma d}{\sqrt{M}} \qquad\qquad \xi = \frac{\sigma\overline{\eta}}{M}. \tag{A.23}
$$

Substituting these results into the above expression,

$$
Y_M \leq \frac{\sigma}{\sqrt{M}}\left(\dim(\mathcal{L}_n)_{n=1}^N + \overline{\eta}\sqrt{2\log\frac{1}{\delta}}\right). \tag{A.24}
$$

□

Proof of Theorem 4.1. Set $\delta \in (0,1)$. Rearranging the results of Lemmas A.1 and A.2, we have that with probability $\geq 1-\delta$,

$$Y_M \leq Y_0 + \min\left(\sqrt{2M\xi^2 \log\frac{1}{\delta}}, \frac{M\tau^2}{\xi} H^{-1}\left(\frac{\xi^2}{M\tau^2}\log\frac{1}{\delta}\right)\right). \tag{A.25}$$

We now apply the results of Lemma A.3(3), where the discrete distribution is specified by atoms $u_n = \mathcal{L}_n/p_n$ with probabilities p_n for $n \in [N]$. Define M_n to be the number of times index n is sampled; then $(M_1, \dots, M_N) \sim \mathsf{Multi}(M, (p_n)_{n=1}^N)$. Then since our vectors are in a Hilbert space, we use Lemma A.3(2) and Lemma A.3(3) to find that

$$Y_M = \left\| \frac{1}{M}\sum_{m=1}^M U_m - \mathbb{E}\left[U\right] \right\| = \left\| \sum_{n=1}^N \frac{M_n}{Mp_n}\mathcal{L}_n - \mathcal{L} \right\| \tag{A.26}$$

$$Y_0 \leq \sqrt{\frac{1}{M}\left(\sum_{n=1}^N \frac{\|\mathcal{L}_n\|^2}{p_n} - \|\mathcal{L}\|^2\right)} \tag{A.27}$$

$$\xi = \frac{1}{M}\max_{m,n}\left\|\frac{\mathcal{L}_n}{p_n} - \frac{\mathcal{L}_m}{p_m}\right\| \tag{A.28}$$

$$\tau^2 = \frac{1}{M^2}\mathbb{E}\left[\|U_m - U_m'\|^2\right] = \frac{2}{M^2}\left(\sum_{n=1}^N \frac{\|\mathcal{L}_n\|^2}{p_n} - \|\mathcal{L}\|^2\right). \tag{A.29}$$

Minimizing both τ^2 and Y_0 over $(p_n)_{n=1}^N$ by setting the derivative to 0 yields

$$p_n = \frac{\|\mathcal{L}_n\|}{\sigma} \qquad\qquad \sigma := \sum_{n=1}^N \|\mathcal{L}_n\|. \tag{A.30}$$

Finally, we have that

$$Y_0 \leq \sqrt{\frac{1}{M}\sigma^2\eta^2} \qquad \tau^2 = \frac{2\sigma^2}{M^2}\left(1 - \frac{\|\mathcal{L}\|^2}{\sigma^2}\right) = \frac{2\sigma^2\eta^2}{M^2} \qquad \xi = \frac{\sigma}{M}\overline{\eta}, \tag{A.31}$$

and $\frac{M_n}{M}\frac{1}{p_n} = W_n$ from Algorithm 1, so

$$\|\mathcal{L}(W) - \mathcal{L}\| \leq \frac{\sigma\eta}{\sqrt{M}} + \min\left(\sqrt{2\frac{\sigma^2\overline{\eta}^2}{M}\log\frac{1}{\delta}}, \frac{2\sigma\eta^2}{\overline{\eta}} H^{-1}\left(\frac{\overline{\eta}^2}{2M\eta^2}\log\frac{1}{\delta}\right)\right) \tag{A.32}$$

$$= \frac{\sigma}{\sqrt{M}}\left(\eta + \eta_M\sqrt{2\log\frac{1}{\delta}}\right) \tag{A.33}$$

$$\eta_M := \min\left(\overline{\eta}, \eta\sqrt{\frac{2M\eta^2}{\overline{\eta}^2\log\frac{1}{\delta}}} H^{-1}\left(\frac{\overline{\eta}^2\log\frac{1}{\delta}}{2M\eta^2}\right)\right). \tag{A.34}$$

□

Lemma A.4. *Given a Hilbert norm, $\overline{\eta}$ from Eq. (3.3) and η from Eq. (4.1) satisfy*

$$\eta \leq \frac{\overline{\eta}}{\sqrt{2}}. \tag{A.35}$$

Proof. Noting $\|\mathcal{L}\|^2 = \langle \mathcal{L}, \mathcal{L} \rangle$ and expanding the definition from Eq. (4.1),

$$\eta^2 = 1 - \sum_{n,m=1}^{N} \frac{\sigma_n \sigma_m}{\sigma^2} \left\langle \frac{\mathcal{L}_n}{\sigma_n}, \frac{\mathcal{L}_m}{\sigma_m} \right\rangle \leq 1 - \min_{n,m \in [N]} \left\langle \frac{\mathcal{L}_n}{\sigma_n}, \frac{\mathcal{L}_m}{\sigma_m} \right\rangle = \frac{1}{2}\overline{\eta}^2 \tag{A.36}$$

where the inequality follows from $\sum_{n,m=1}^{N} \frac{\sigma_n \sigma_m}{\sigma^2} = 1$. □

Lemma A.5. *$\mathcal{L}$ is in the relative interior of the convex hull of $\left\{ \frac{\sigma}{\|\mathcal{L}_n\|} \mathcal{L}_n \right\}_{n=1}^{N}$.*

Proof. First, since K (the kernel matrix of inner products defined in Eq. (4.6)) is a symmetric postive semidefinite $N \times N$ matrix, there exists an $N \times N$ matrix U such that $\|\mathcal{L}(w)\|^2 = w^T K w = w^T U^T U w = \|u(w)\|_2^2$, where $u(w) := \sum_{n=1}^{N} w_n u_n$, $u := \sum_{n=1}^{N} u_n$, and $u_n \in \mathbb{R}^N$ are the columns of U. Therefore the mapping $\mathcal{L}(w) \to u(w)$ is a linear isometry from the Hilbert space to $\mathbb{R}^N$, so if u is in the relative interior of the convex hull of $\left\{ \frac{\sigma}{\|u_n\|} u_n \right\}_{n=1}^{N}$, the result follows. Let y be any other point in the convex hull in $\mathbb{R}^N$, with coefficients γ_n. If we set

$$\lambda = \min_{n : \gamma_n > \frac{\|u_n\|}{\sigma}} \frac{\gamma_n}{\gamma_n - \frac{\|u_n\|}{\sigma}} \tag{A.37}$$

where the minimum of an empty set is defined to be ∞, then $\lambda u + (1-\lambda) y$ is in the convex hull and $\lambda > 1$. Since for any point y we can find such a λ, the result follows from (Rockafeller, 1970, Theorem 6.4, p. 47). □

Lemma A.6. *The logistic equation,*

$$x_{n+1} = \alpha x_n (1 - x_n), \tag{A.38}$$

for $x_0, \alpha \in [0,1]$ satisfies

$$\forall n \in \mathbb{N}, \quad x_n \leq \frac{x_0}{\alpha^{-n} + x_0 n}. \tag{A.39}$$

Proof. The proof proceeds by induction. The bound holds at $n = 0$ and $n = 1$ since

$$x_0 \leq \frac{x_0}{\alpha^0 + 0} = x_0 \tag{A.40}$$

$$x_1 \leq \alpha x_0 (1 - x_0) \leq \alpha \frac{x_0}{1 + x_0} = \frac{x_0}{\alpha^{-1} + \alpha^{-1} x_0} \leq \frac{x_0}{\alpha^{-1} + x_0 \cdot 1}. \tag{A.41}$$

Next, assume that the bound holds at $n \geq 2$. Note that for any $n \geq 2$,

$$\max_{x \in [0,1]} \alpha x (1 - x) = \alpha \frac{1}{4} \leq \frac{1}{2} \implies x_n \leq \frac{1}{2}, \tag{A.42}$$

and

$$\frac{x_0}{\alpha^{-n} + x_0 n} \leq \frac{x_0}{\alpha^{-2} + 2 x_0} \leq \frac{1}{2}. \tag{A.43}$$

Combined with the fact that $\alpha x (1 - x)$ is increasing for all $x \in [0, 1/2]$, we have that

$$x_{n+1} = \alpha x_n (1 - x_n) \leq \alpha \frac{x_0}{\alpha^{-n} + x_0 n} \left(1 - \frac{x_0}{\alpha^{-n} + x_0 n} \right) \tag{A.44}$$

$$= \frac{x_0}{\alpha^{-n} + x_0 n} \left(\frac{\alpha^{-n} + x_0 n - x_0}{\alpha^{-(n+1)} + \alpha^{-1} x_0 n} \right) \tag{A.45}$$

$$\leq \frac{x_0}{\alpha^{-n} + x_0 n} \frac{\alpha^{-n} + x_0 n}{\alpha^{-(n+1)} + x_0 (n+1)} \tag{A.46}$$

$$= \frac{x_0}{\alpha^{-(n+1)} + x_0(n+1)}. \tag{A.47}$$

□

Proof of Lemma 4.3. Let w_t be the weight vector at iteration t in Algorithm 2, and let f_t and d_t be the Frank-Wolfe vertex index and direction, respectively, from Eq. (4.8). For brevity, denote the cost $J(w) := (w-1)^T K(w-1)$. For any $\gamma \in \mathbb{R}$, if we let $w_{t+1} = w_t + \gamma d_t$ we have that

$$J(w_{t+1}) = J(w_t) + 2\gamma d_t^T K(w_t - 1) + \gamma^2 d_t^T K d_t. \tag{A.48}$$

Minimizing Eq. (A.48) over $\gamma \in \mathbb{R}$ yields Eq. (4.9) (expressed as a quadratic form with gram matrix K),

$$\gamma_t = \frac{d_t^T K(1 - w_t)}{d_t^T K d_t}. \tag{A.49}$$

Suppose $\gamma_t < 0$. Then $d_t^T K(1 - w_t) < 0$; but d_t maximizes this product over feasible directions, so

$$0 > d_t^T K(1 - w_t) > (1 - w_t)^T K(1 - w_t) = J(w_t) \geq 0, \tag{A.50}$$

which is a contradiction. Now suppose $\gamma_t > 1$. Then

$$d_t^T K(1 - w_t) > d_t^T K d_t, \tag{A.51}$$

and Eq. (A.50) holds again, so if we were to select $\gamma = 1$ in Eq. (A.48), we would have

$$0 \leq J(w_{t+1}) < J(w_t) + d_t^T K(w_t - 1) \leq 0, \tag{A.52}$$

which is another contradiction, so $\gamma_t \leq 1$. Therefore $\gamma_t \in [0, 1]$ □

Proof of Theorem 4.4. Using the same notation as the proof of Lemma 4.3 above, first note that $J(w_0) \leq \sigma^2 \eta^2$ as initialized by Eq. (4.7): for any $\xi \in \mathbb{R}_+^N$ with $\sum_n \xi_n = 1$,

$$\frac{J(w_0)}{\sigma^2} = 1 - 2\left\langle \frac{\mathcal{L}_{f_0}}{\sigma_{f_0}}, \frac{\mathcal{L}}{\sigma} \right\rangle + \frac{\|\mathcal{L}\|^2}{\sigma^2} \leq 1 - 2\sum_{n=1}^N \xi_n \left\langle \frac{\mathcal{L}_n}{\sigma_n}, \frac{\mathcal{L}}{\sigma} \right\rangle + \frac{\|\mathcal{L}\|^2}{\sigma^2} \tag{A.53}$$

since f_0 maximizes $\langle \mathcal{L}, \mathcal{L}_n/\sigma_n \rangle$ over $n \in [N]$, and picking $\xi_n = \sigma_n/\sigma$ yields

$$\frac{J(w_0)}{\sigma^2} \leq 1 - 2\sum_{n=1}^N \left\langle \frac{\mathcal{L}_n}{\sigma}, \frac{\mathcal{L}}{\sigma} \right\rangle + \frac{\|\mathcal{L}\|^2}{\sigma^2} = 1 - \frac{\|\mathcal{L}\|^2}{\sigma^2} = \eta^2. \tag{A.54}$$

By Lemma 4.3, we are guaranteed that each Frank-Wolfe iterate using exact line search is feasible, and substituting Eq. (A.49) into Eq. (A.48) yields

$$J(w_{t+1}) = J(w_t) - \frac{\left(d_t^T K(1 - w_t)\right)^2}{d_t^T K d_t} \tag{A.55}$$

$$= J(w_t)\left(1 - \left\langle \frac{\frac{\sigma}{\sigma_{f_t}}\mathcal{L}_{f_t} - \mathcal{L}(w_t)}{\|\frac{\sigma}{\sigma_{f_t}}\mathcal{L}_{f_t} - \mathcal{L}(w_t)\|}, \frac{\mathcal{L} - \mathcal{L}(w_t)}{\|\mathcal{L} - \mathcal{L}(w_t)\|} \right\rangle^2\right). \tag{A.56}$$

We now employ a technique due to Guélat and Marcotte (1986): by Lemma A.5, $\mathcal{L}$ is in the relative interior of the convex hull of the $\left\{\frac{\sigma}{\sigma_n}\mathcal{L}_n\right\}_{n=1}^N$, so there exists an $r > 0$ such that for any feasible w,

$$\mathcal{L}(w) + (\|\mathcal{L} - \mathcal{L}(w)\| + r)\frac{\mathcal{L} - \mathcal{L}(w)}{\|\mathcal{L} - \mathcal{L}(w)\|} \tag{A.57}$$

is also in the convex hull. Thus, since the Frank-Wolfe vertex $\frac{\sigma}{\sigma_{f_t}}\mathcal{L}_{f_t}$ maximizes $\langle \mathcal{L}(w) - \mathcal{L}(w_t), \mathcal{L} - \mathcal{L}(w_t)\rangle$ over feasible w, we have that

$$\left\langle \frac{\frac{\sigma}{\sigma_{f_t}}\mathcal{L}_{f_t} - \mathcal{L}(w_t)}{\|\frac{\sigma}{\sigma_{f_t}}\mathcal{L}_{f_t} - \mathcal{L}(w_t)\|}, \frac{\mathcal{L} - \mathcal{L}(w_t)}{\|\mathcal{L} - \mathcal{L}(w_t)\|} \right\rangle \geq \left\langle \frac{(\|\mathcal{L} - \mathcal{L}(w_t)\| + r)\frac{\mathcal{L}-\mathcal{L}(w_t)}{\|\mathcal{L}-\mathcal{L}(w_t)\|}}{\|\frac{\sigma}{\sigma_{f_t}}\mathcal{L}_{f_t} - \mathcal{L}(w_t)\|}, \frac{\mathcal{L} - \mathcal{L}(w_t)}{\|\mathcal{L} - \mathcal{L}(w_t)\|} \right\rangle \tag{A.58}$$

$$= \frac{\sqrt{J(w_t)} + r}{\|\frac{\sigma}{\sigma_{f_t}}\mathcal{L}_{f_t} - \mathcal{L}(w_t)\|} \tag{A.59}$$

$$\geq \frac{\sqrt{J(w_t)} + r}{\sigma\overline{\eta}}. \tag{A.60}$$

Substituting this into Eq. (A.56) yields

$$J(w_{t+1}) \leq J(w_t)\left(1 - \left(\frac{\sqrt{J(w_t)} + r}{\sigma\overline{\eta}}\right)^2\right) \leq J(w_t)\left(\nu^2 - \frac{J(w_t)}{\sigma^2\overline{\eta}^2}\right), \tag{A.61}$$

where $\nu := 1 - \frac{r^2}{\sigma^2\overline{\eta}^2}$. Defining $x_t := \frac{J(w_t)}{\sigma^2\overline{\eta}^2\nu^2}$, we have that $0 \leq x_t \leq 1$ and

$$x_{t+1} \leq \nu^2 x_t(1 - x_t), \tag{A.62}$$

and so Lemma A.6 implies that

$$\frac{J(w_t)}{\sigma^2\overline{\eta}^2\nu^2} \leq \frac{\frac{J(w_0)}{\sigma^2\overline{\eta}^2\nu^2}}{\nu^{-2t} + \frac{J(w_0)}{\sigma^2\overline{\eta}^2\nu^2}t}. \tag{A.63}$$

Further, since the function $\frac{a}{a+b}$ is monotonically increasing in a for all $a, b \geq 0$, we can use the bound on the initial objective $J(w_0)$, yielding

$$J(w_t) \leq \frac{\sigma^2\eta^2\overline{\eta}^2\nu^2}{\overline{\eta}^2\nu^{2-2t} + \eta^2 t} \tag{A.64}$$

The proof concludes by noting that we compute $M - 1$ iterations after initialization to construct a coreset of size $\leq M$. The second stated bound results from the fact that $\overline{\eta} \geq \eta$ and $\nu \leq 1$.

The weaker bound in the note after the theorem is a result of a technique very similar to that commonly found in past work (Clarkson, 2010; Jaggi, 2013): starting from Eq. (A.48), we bound $d_t^T K d_t \leq \sigma^2\overline{\eta}^2$ and $d_t^T K(w_t - 1) \leq -J(w_t)$, and then use recursion to prove that $J(w_t) \leq \frac{4\sigma^2\overline{\eta}^2}{3t+4}$ given $\gamma_t = \frac{2}{3t+4}$. □

Proof of Theorem 5.2. Suppose $\max_{m,n} \left|\langle \mathcal{L}_n, \mathcal{L}_m\rangle - v_n^T v_m\right| \leq \epsilon$. Then

$$(w-1)^T K(w-1) - (w-1)^T V(w-1) \leq \sum_{m,n} |w_n - 1|\,|w_m - 1|\left|\langle \mathcal{L}_n, \mathcal{L}_m\rangle - v_n^T v_m\right| \leq \|w-1\|_1^2\,\epsilon. \tag{A.65}$$

We now bound the probability that the above inequality holds, assuming $D\nabla\mathcal{L}_n(\mu)_d\nabla\mathcal{L}_m(\mu)_d$ (when using $\mathcal{D}_{\hat{\pi},F}$) or $\mathcal{L}_n(\mu)\mathcal{L}_m(\mu)$ (when using $\mathcal{D}_{\hat{\pi},2}$) is sub-Gaussian with constant ξ^2. For brevity denote the true vector $\mathcal{L}_n$ as $\mathcal{L}_n$ and its random projection as v_n. Then

$$\begin{aligned}
&\mathbb{P}\left(\max_{m,n}\left|\langle\mathcal{L}_n,\mathcal{L}_m\rangle - v_n^T v_m\right| \geq \epsilon\right) \\
\leq &\sum_{m,n}\mathbb{P}\left(\left|\langle\mathcal{L}_n,\mathcal{L}_m\rangle - v_n^T v_m\right| \geq \epsilon\right) && \text{(A.66)}\\
\leq &N^2\max_{m,n}\mathbb{P}\left(\left|\langle\mathcal{L}_n,\mathcal{L}_m\rangle - v_n^T v_m\right| \geq \epsilon\right) && \text{(A.67)}\\
= &N^2\max_{m,n}\mathbb{P}\left(\left|\langle\mathcal{L}_n,\mathcal{L}_m\rangle - \frac{1}{J}\sum_{j=1}^{J} v_{nj}v_{mj}\right| \geq \epsilon\right) && \text{(A.68)}\\
\leq &2N^2 e^{-\frac{J\epsilon^2}{2\xi^2}}, && \text{(A.69)}
\end{aligned}$$

using Hoeffding's inequality for sub-Gaussian variables. Thus if we fix $\delta \in (0,1)$, with probability $\geq 1-\delta$,

$$\sqrt{\frac{2\xi^2}{J}\log\frac{2N^2}{\delta}} \geq \epsilon. \tag{A.70}$$

Therefore, with probability $\geq 1-\delta$,

$$\|\mathcal{L}(w)-\mathcal{L}\|^2 \leq \|v(w)-v\|^2 + \|w-1\|_1^2\sqrt{\frac{2\xi^2}{J}\log\frac{2N^2}{\delta}}. \tag{A.71}$$

□

Appendix B. Additional Results

This section contains supplementary quantitative evaluations. Fig. 7 compares Hilbert importance sampling to uniformly random subsampling and uniform coresets. These results demonstrate that all subsampling techniques perform similarly, with Hilbert coresets often the best choice of the three. The Frank–Wolfe constructions outperform subsampling techniques across all models and datasets considered. Figs. 8 and 9 compare the weighted 2-norm from Eq. (5.3) to the weighted Fisher information norm from Eq. (5.1) in both importance sampling and Frank–Wolfe-based Hilbert coreset constructions. These results show that the 2-norm and F-norm perform similarly in all cases.

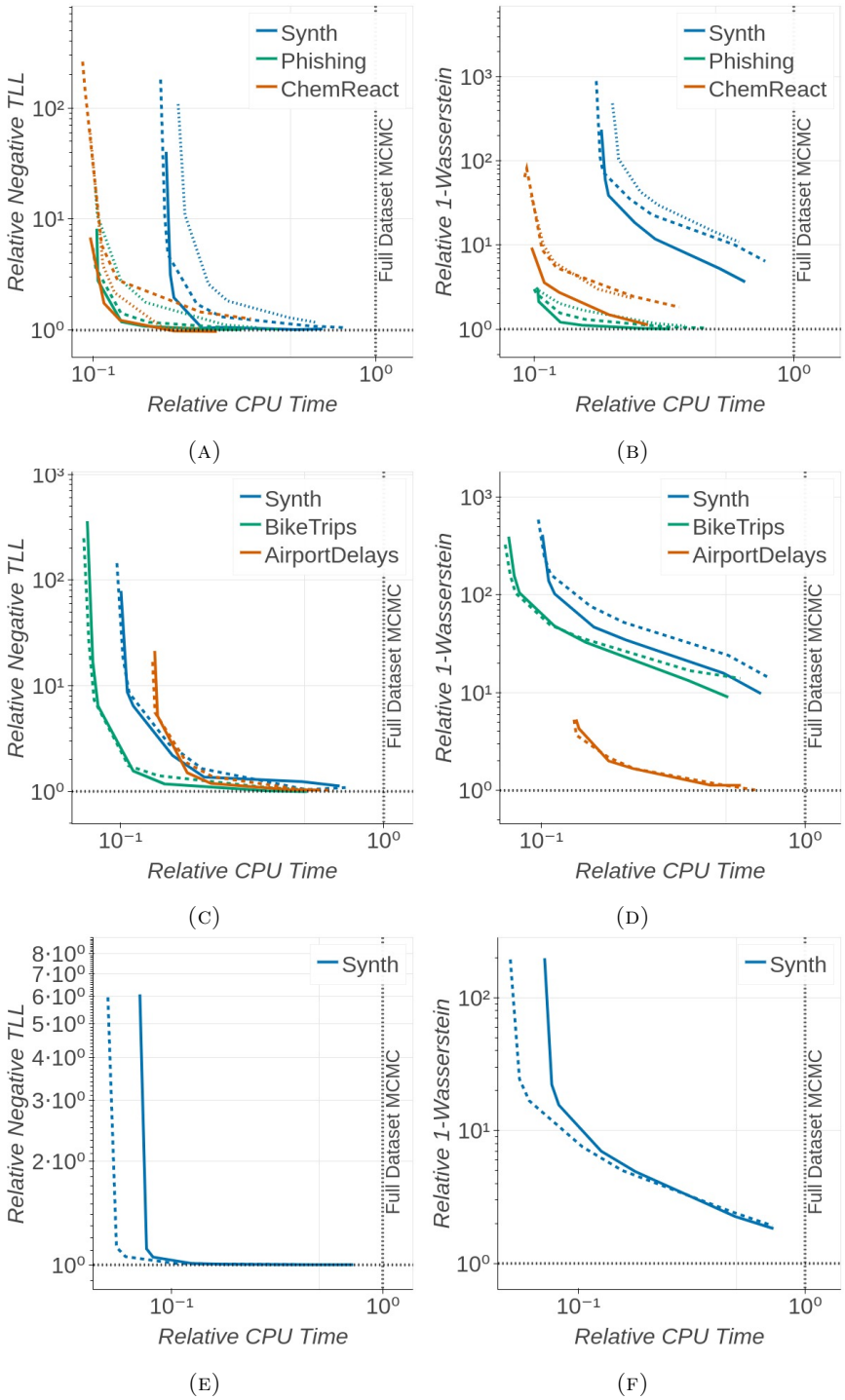


FIGURE 7. Comparisons for IS-F (solid), uniform coresets (dotted), and uniform random subsampling (dashed) on (7a, 7b) logistic regression, (7c, 7d) Poisson regression, and (7e, 7f) directional clustering. Both axes are normalized; see Section 7.3.

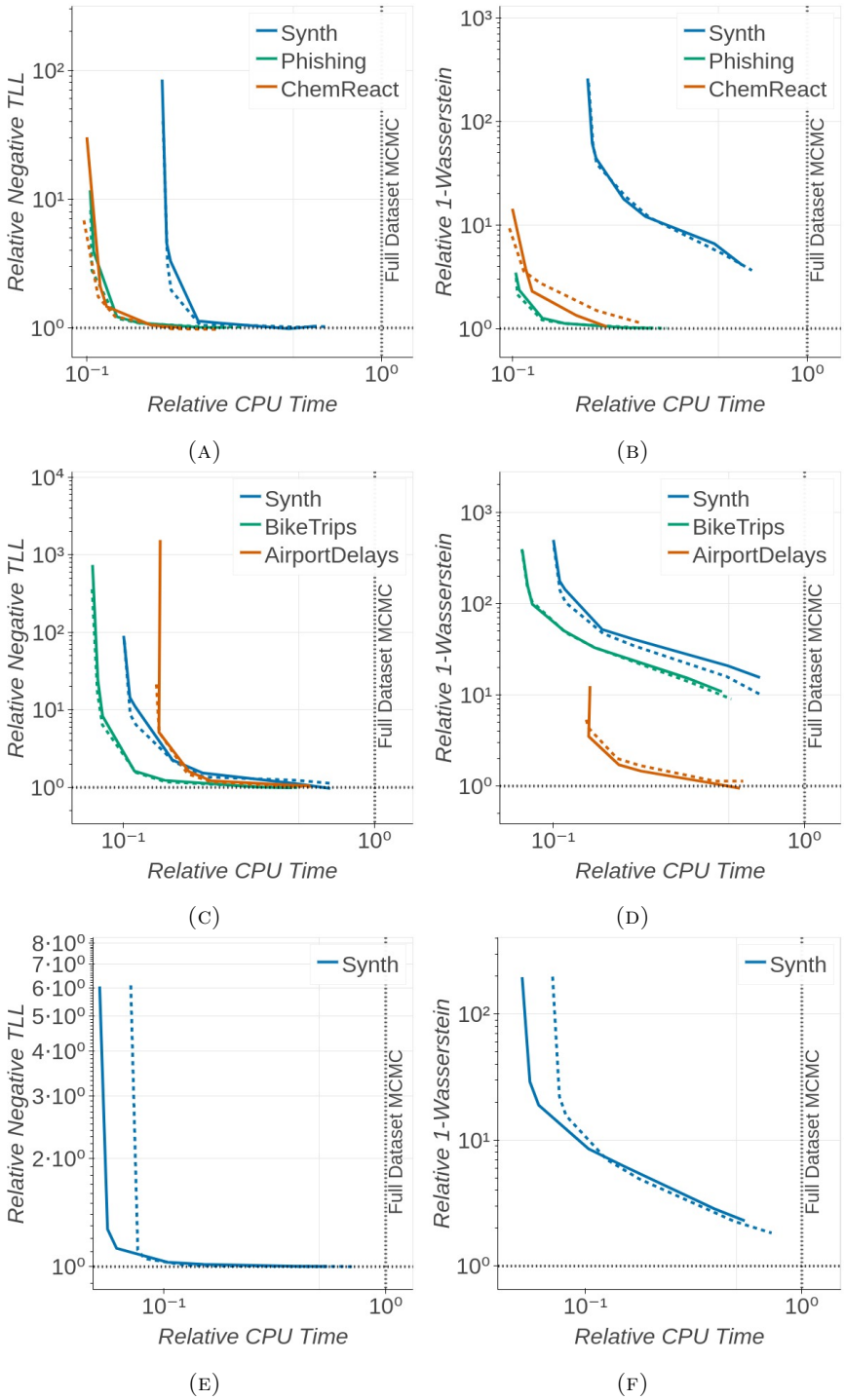


FIGURE 8. Comparisons for IS-2 (solid) and IS-F (dashed) on (8a, 8b) logistic regression, (8c, 8d) Poisson regression, and (8e, 8f) directional clustering. Both axes are normalized; see Section 7.3.

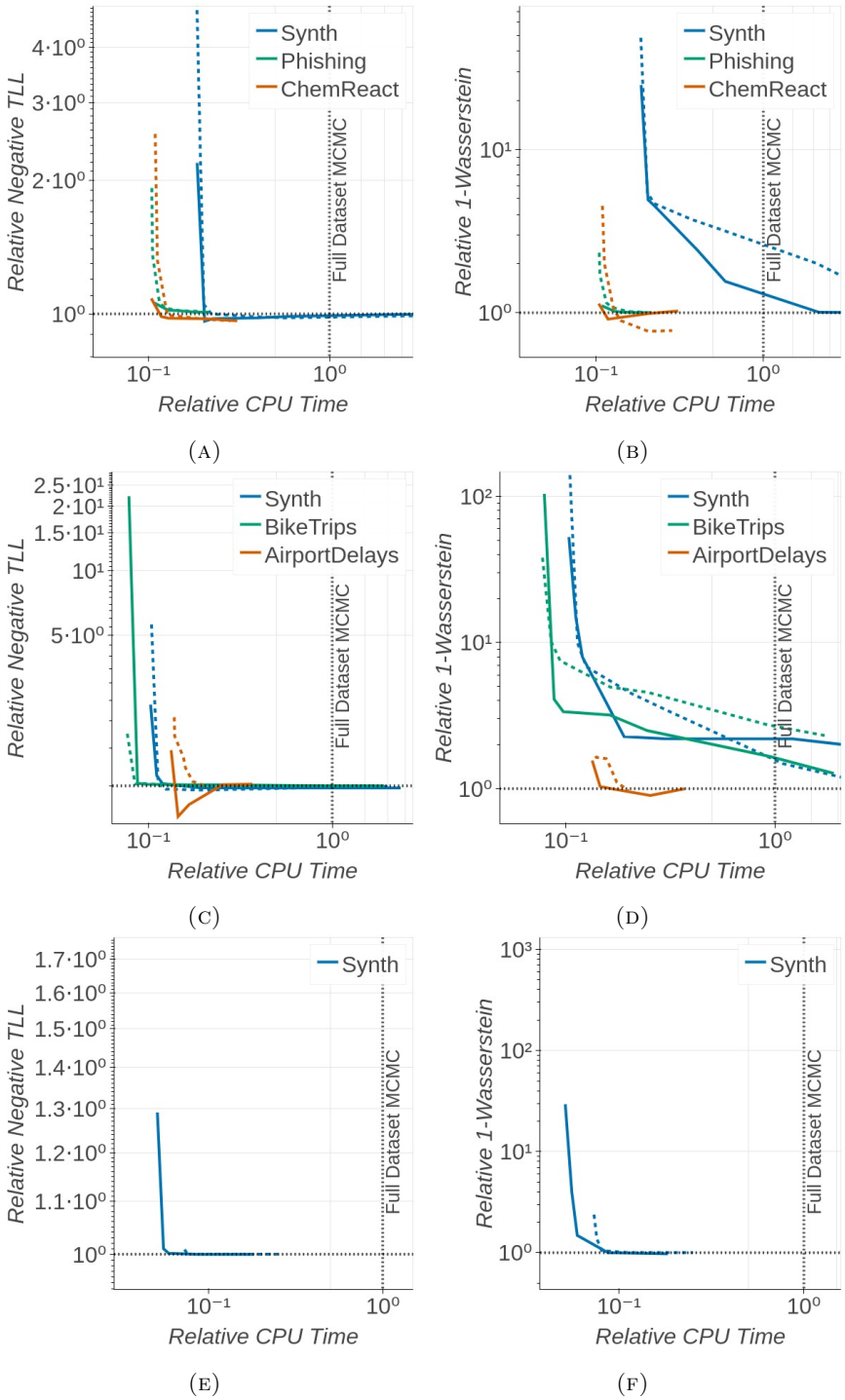


FIGURE 9. Comparisons for FW-2 (solid) and FW-F (dashed) on (9a, 9b) logistic regression, (9c, 9d) Poisson regression, and (9e, 9f) directional clustering. Both axes are normalized; see Section 7.3.

Appendix C. Derivation of the Gaussian uniform coreset sensitivity

The sensitivity of observation y_n used in the construction of a Bayesian coreset (Huggins et al., 2016) (ignoring constants) is

$$N\sigma_n = \sup_{\mu\in\mathbb{R}^d} \frac{N\mathcal{L}_n(\mu)}{\mathcal{L}(\mu)} = \sup_{\mu\in\mathbb{R}^d} \frac{N\left(y_n-\mu\right)^T\left(y_n-\mu\right)}{\sum_{m=1}^N \left(y_m-\mu\right)^T\left(y_m-\mu\right)}. \tag{C.1}$$

By noting that

$$\frac{1}{N}\sum_{m=1}^N \left(y_m-\mu\right)^T\left(y_m-\mu\right) = \frac{1}{N}\sum_{m=1}^N y_m^T y_m - \bar{y}^T\bar{y} + \left(\mu-\bar{y}\right)^T\left(\mu-\bar{y}\right), \tag{C.2}$$

where $\bar{y} := \frac{1}{N}\sum_{m=1}^N y_m$, we can keep the denominator constant by varying μ on the ball centered at $\bar{y}$ of constant radius. The maximum of the numerator while keeping the denominator constant happens when μ lies on the 1d affine space between $\bar{y}$ and y_n; so we can reparametrize $\mu = \lambda\bar{y} + (1-\lambda)y_n$ for $\lambda\in\mathbb{R}$, yielding the optimization

$$\sup_{\mu\in\mathbb{R}^d} \frac{N\mathcal{L}_n(\mu)}{\mathcal{L}(\mu)} = \sup_{\lambda\in\mathbb{R}} \frac{\lambda^2\left(y_n-\bar{y}\right)^T\left(y_n-\bar{y}\right)}{\frac{1}{N}\sum_{m=1}^N y_m^T y_m - \bar{y}^T\bar{y} + (1-\lambda)^2\left(y_n-\bar{y}\right)^T\left(y_n-\bar{y}\right)} \tag{C.3}$$

for which the optimum occurs at $\lambda^\star = \left(\frac{\frac{1}{N}\sum_{m=1}^N y_m^T y_m - \bar{y}^T\bar{y}}{\left(y_n-\bar{y}\right)^T\left(y_n-\bar{y}\right)} + 1\right)$ with value

$$N\sigma_n = \sup_{\mu\in\mathbb{R}^d} \frac{N\mathcal{L}_n(\mu)}{\mathcal{L}(\mu)} = 1 + \frac{\left(y_n-\bar{y}\right)^T\left(y_n-\bar{y}\right)}{\frac{1}{N}\sum_{m=1}^N y_m^T y_m - \bar{y}^T\bar{y}}. \tag{C.4}$$

References

Agarwal, P. K., Har-Peled, S., and Varadarajan, K. R. (2005). "Geometric approximation via coresets." *Combinatorial and computational geometry*, 52: 1–30.

Ahfock, D., Astle, W., and Richardson, S. (2017). "Statistical properties of sketching algorithms." *arXiv:1706.03665*.

Ahn, S., Korattikara, A., and Welling, M. (2012). "Bayesian posterior sampling via stochastic gradient Fisher scoring." In *International Conference on Machine Learning*.

Bachem, O., Lucic, M., Hassani, S. H., and Krause, A. (2016). "Approximate k-means++ in sublinear time." In *AAAI Conference on Artificial Intelligence*.

Bachem, O., Lucic, M., and Krause, A. (2015). "Coresets for nonparametric estimation—the case of DP-means." In *International Conference on Machine Learning*.

Banerjee, A., Dhillon, I., Ghosh, J., and Sra, S. (2005). "Clustering on the unit hypersphere using von Mises-Fisher distributions." *Journal of Machine Learning Research*, 6: 1345–1382.

Bardenet, R., Doucet, A., and Holmes, C. (2015). "On Markov chain Monte Carlo methods for tall data." *arXiv:1505.02827*.

Bardenet, R., Doucet, A., and Holmes, C. C. (2014). "Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach." In *International Conference on Machine Learning*, 405–413.

Bardenet, R. and Maillard, O.-A. (2015). "A note on replacing uniform subsampling by random projections in MCMC for linear regression of tall datasets." Technical report, HAL Id: hal-01248841.

Bishop, C. (2006). *Pattern recognition and machine learning*. Springer.

Boucheron, S., Lugosi, G., and Massart, P. (2013). *Concentration inequalities: a nonasymptotic theory of independence*. Oxford University Press.

Braverman, V., Feldman, D., and Lang, H. (2016). "New frameworks for offline and streaming coreset constructions." *arXiv:1612.00889*.

Broderick, T., Boyd, N., Wibisono, A., Wilson, A., and Jordan, M. (2013). "Streaming variational Bayes." In *Advances in Neural Information Processing Systems*.

Campbell, T., Straub, J., Fisher III, J. W., and How, J. (2015). "Streaming, distributed variational inference for Bayesian nonparametrics." In *Advances in Neural Information Processing Systems*.

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

Entezari, R., Craiu, R., and Rosenthal, J. (2016). "Likelihood inflating sampling algorithm." *arXiv:1605.02113*.

Feldman, D., Faulkner, M., and Krause, A. (2011). "Scalable training of mixture models via coresets." In *Advances in Neural Information Processing Systems*, 2142–2150.

Feldman, D. and Langberg, M. (2011). "A unified framework for approximating and clustering data." In *Proceedings of the* 43^{rd} *Annual ACM Symposium on Theory of Computing*, 569–578.

Feldman, D., Schmidt, M., and Sohler, C. (2013). "Turning big data into tiny data: constant-size coresets for k-means, PCA and projective clustering." In *Proceedings of the* 24^{th} *Annual ACM–SIAM Symposium on Discrete Algorithms*, 1434–1453.

Frank, M. and Wolfe, P. (1956). "An algorithm for quadratic programming." *Naval Research Logistics Quarterly*, 3: 95–110.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D. (2013). *Bayesian data analysis*. CRC Press, 3rd edition.

Geppert, L., Ickstadt, K., Munteanu, A., Quedenfeld, J., and Sohler, C. (2017). "Random projections for Bayesian regression." *Statistics and Computing*, 27: 79–101.

Guélat, J. and Marcotte, P. (1986). "Some comments on Wolfe's 'away step'." *Mathematical Programming*, 35: 110–119.

Han, L., Yang, T., and Zhang, T. (2016). "Local uncertainty sampling for large-scale multi-class logistic regression." *arXiv:1604.08098*.

Hoffman, M. and Gelman, A. (2014). "The No-U-Turn Sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." *Journal of Machine Learning Research*, 15: 1351–1381.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). "Stochastic variational inference." *Journal of Machine Learning Research*, 14: 1303–1347.

Huggins, J., Adams, R., and Broderick, T. (2017). "PASS-GLM: polynomial approximate sufficient statistics for scalable Bayesian GLM inference." In *Advances in Neural Information Processing Systems*.

Huggins, J., Campbell, T., and Broderick, T. (2016). "Coresets for Bayesian logistic regression." In *Advances in Neural Information Processing Systems*.

Jaggi, M. (2013). "Revisiting Frank-Wolfe: projection-free sparse convex optimization." In *International Conference on Machine Learning*.

Johnson, O. and Barron, A. (2004). "Fisher information inequalities and the central limit theorem." *Probability Theory and Related Fields*, 129: 391–409.

Jordan, M., Ghahramani, Z., Jaakkola, T., and Saul, L. (1999). "An introduction to variational methods for graphical models." *Machine Learning*, 37: 183–233.

Korattikara, A., Chen, Y., and Welling, M. (2014). "Austerity in MCMC land: cutting the Metropolis-Hastings budget." In *International Conference on Machine Learning*.

Kucukelbir, A., Ranganath, R., Gelman, A., and Blei, D. (2015). "Automatic variational inference in Stan." In *Advances in Neural Information Processing Systems*.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. (2017). "Automatic differentiation variational inference." *Journal of Machine Learning Research*, 18: 1–45.

Lacoste-Julien, S. and Jaggi, M. (2015). "On the global linear convergence of Frank-Wolfe optimization variants." In *Advances in Neural Information Processing Systems*.

Langberg, M. and Schulman, L. (2010). "Universal ϵ-approximators for integrals." In *Proceedings of the* 21^{st} *Annual ACM–SIAM Symposium on Discrete Algorithms*, 598–607.

Lucic, M., Bachem, O., and Krause, A. (2016). "Strong coresets for hard and soft bregman clustering with applications to exponential family mixtures." In *International Conference on Artificial Intelligence and Statistics*.

Maclaurin, D. and Adams, R. (2014). "Firefly Monte Carlo: exact MCMC with subsets of data." In *Conference on Uncertainty in Artificial Intelligence*.

Neal, R. (2011). "MCMC using Hamiltonian dynamics." In Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (eds.), *Handbook of Markov chain Monte Carlo*, chapter 5. CRC Press.

Rabinovich, M., Angelino, E., and Jordan, M. I. (2015). "Variational consensus Monte Carlo." *arXiv:1506.03074*.

Rahimi, A. and Recht, B. (2007). "Random features for large-scale kernel machines." In *Advances in Neural Information Processing Systems*.

Ranganath, R., Gerrish, S., and Blei, D. (2014). "Black box variational inference." In *International Conference on Artificial Intelligence and Statistics*.

Rockafeller, R. (1970). *Convex analysis*. Princeton University Press.

Scott, S., Blocker, A., Bonassi, F., Chipman, H., George, E., and McCulloch, R. (2016). "Bayes and big data: the consensus Monte Carlo algorithm." *International Journal of Management Science and Engineering Management*, 11: 78–88.

Srivastava, S., Cevher, V., Tran-Dinh, Q., and Dunson, D. (2015). "WASP: scalable Bayes via barycenters of subset posteriors." In *International Conference on Artificial Intelligence and Statistics*.

Wainwright, M. and Jordan, M. (2008). "Graphical models, exponential families, and variational inference." *Foundations and Trends in Machine Learning*, 1(1–2): 1–305.

Welling, M. and Teh, Y. W. (2011). "Bayesian learning via stochastic gradient Langevin dynamics." In *International Conference on Machine Learning*.

Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology
URL: http://www.trevorcampbell.me/
E-mail address: tdjc@mit.edu

Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology
URL: http://www.tamarabroderick.com
E-mail address: tbroderick@csail.mit.edu