Abstract

The proliferation of automated inference algorithms in Bayesian statistics has provided practitioners newfound access to fast, reproducible data analysis and powerful statistical models. Designing automated methods that are also both computationally scalable and theoretically sound, however, remains a significant challenge. Recent work on Bayesian coresets takes the approach of compressing the dataset before running a standard inference algorithm, providing both scalability and guarantees on posterior approximation error. But the automation of past coreset methods is limited because they depend on the availability of a reasonable coarse posterior approximation, which is difficult to specify in practice. In the present work we remove this requirement by formulating coreset construction as sparsity-constrained variational inference within an exponential family. This perspective leads to a novel construction via greedy optimization, and also provides a unifying information-geometric view of present and past methods. The proposed Riemannian coreset construction algorithm is fully automated, requiring no problem-specific inputs aside from the probabilistic model and dataset. In addition to being significantly easier to use than past methods, experiments demonstrate that past coreset constructions are fundamentally limited by the fixed coarse posterior approximation; in contrast, the proposed algorithm is able to continually improve the coreset, providing state-of-the-art Bayesian dataset summarization with orders-of-magnitude reduction in KL divergence to the exact posterior.

1 Introduction

Bayesian statistical models are powerful tools for learning from data, with the ability to encode complex hierarchical dependence and domain expertise, as well as coherently quantify uncertainty in latent parameters. In practice, however, exact Bayesian inference is typically intractable, and we must use approximate inference algorithms such as Markov chain Monte Carlo (MCMC) [1,2] and variational inference (VI) [3,4]. Until recently, implementations of these methods were created on a per-model basis, requiring expert input to design the MCMC transition kernels or derive VI gradient updates. But developments in automated tools—e.g., automatic differentiation [5,6], “black-box” gradient estimates [7], and Hamiltonian transition kernels [8,9]—have obviated much of this expert input, greatly expanding the repertoire of Bayesian models accessible to practitioners.

In modern data analysis problems, automation alone is insufficient; inference algorithms must also be computationally scalable—to handle the ever-growing size of datasets—and provide theoretical guarantees on the quality of their output such that statistical practitioners may confidently use them in failure-sensitive settings. Here the standard set of tools falls short. Designing correct MCMC schemes in the large-scale data setting is a challenging, problem-specific task [10,12]; and despite
recent results in asymptotic theory [13][16], it is difficult to assess the effect of the variational family on VI approximations for finite data, where a poor choice can result in severe underestimation of posterior uncertainty [17 Ch. 21]. Other scalable Bayesian inference algorithms have largely been developed by modifying standard inference algorithms to handle distributed or streaming data processing [10][11][18][29], which tend to have no guarantees on inferential quality and require extensive model-specific expert tuning.

Bayesian coresets (“core of a dataset”) [30][32] are an alternative approach—based on the notion that large datasets often contain a significant fraction of redundant data—that summarize and sparsify the data as a preprocessing step before running a standard inference algorithm such as MCMC or VI. In contrast to other large-scale inference techniques, Bayesian coreset construction is computationally inexpensive, simple to implement, and provides theoretical guarantees relating coreset size to posterior approximation quality. However, state-of-the-art algorithms formulate coreset construction as a sparse regression problem in a Hilbert space, which involves the choice of a weighted $L^2$ inner product [31]. If left to the user, the choice of weighting distribution significantly reduces the overall automation of the approach; and current methods for finding the weighting distribution programatically are generally as expensive as posterior inference on the full dataset itself. Further, even if an appropriate inner product is specified, computing it exactly is typically intractable, requiring the use of finite-dimensional projections for approximation [31]. Although the problem in finite-dimensions can be studied using well-known techniques from sparse regression, compressed sensing, random sketching, boosting, and greedy approximation [33][51], these projections incur an unknown error in the construction process in practice, and preclude asymptotic consistency as the coreset size grows.

In this work, we provide a new formulation of coreset construction as exponential family variational inference with a sparsity constraint. The fact that coresets form a sparse subset of an exponential family is crucial in two regards. First, it enables tractable unbiased Kullback-Leibler (KL) divergence gradient estimation, which is used in the development of a novel coreset construction algorithm based on greedy optimization. In contrast to past work, this algorithm is fully automated, with no problem-specific inputs aside from the probabilistic model and dataset. Second, it provides a unifying view and strong theoretical underpinnings of both the present and past coreset constructions through Riemannian information geometry. In particular, past methods are shown to operate in a single tangent space of the coreset manifold; our experiments show that this fundamentally limits the quality of the coreset constructed with these methods. In contrast, the proposed method proceeds along the manifold towards the posterior target, and is able to continually improve its approximation. Furthermore, new relationships between the optimization objective of past approaches and the coreset posterior KL divergence are derived. The paper concludes with experiments demonstrating that, compared with past methods, Riemannian coreset construction is both easier to use and provides orders-of-magnitude reduction in KL divergence to the exact posterior.

2 Background

In the problem setting of the present paper, we are given a probability density $\pi(\theta)$ for variables $\theta \in \Theta$ that decomposes into $N$ potentials $(f_n(\theta))_{n=1}^N$ and a base density $\pi_0(\theta),$

$$\pi(\theta) := \frac{1}{Z} \exp \left( \sum_{n=1}^{N} f_n(\theta) \right) \pi_0(\theta), \quad (1)$$

where $Z$ is the (unknown) normalization constant. Such distributions arise frequently in a number of scenarios: for example, in Bayesian statistical inference problems with conditionally independent data given $\theta$, the functions $f_n$ are the log-likelihood terms for the $N$ data points, $\pi_0$ is the prior density, and $\pi$ is the posterior; or in undirected graphical models, the functions $f_n$ and $\log \pi_0$ might represent $N + 1$ potentials. The algorithms and analysis in the present work are agnostic to their particular meaning, but for clarity we will focus on the setting of Bayesian inference throughout.

As it is often intractable to compute expectations under $\pi$ exactly, practitioners have turned to approximate algorithms. Markov chain Monte Carlo (MCMC) methods [1][8][9], which return approximate samples from $\pi$, remain the gold standard for this purpose. But since each sample typically requires at least one evaluation of a function proportional to $\pi$ with computational cost $O(N)$, in the large $N$ setting it is expensive to obtain sufficiently many samples to provide high confidence in empirical estimates. To reduce the cost of MCMC, we can instead run it on a small, weighted subset
of data known as a Bayesian coreset [30], a concept originating from the computational geometry and optimization literature [32, 57]. Let \( w \in \mathbb{R}_{\geq 0}^{N} \) be a sparse vector of nonnegative weights such that only \( M \ll N \) are nonzero, i.e. \( \| w \|_0 \equiv \sum_{n=1}^{N} 1[w_n > 0] \leq M \). Then we approximate the full log-density with a \( w \)-reweighted sum with normalization \( Z(w) > 0 \) and run MCMC on the approximation \( \pi_w(\theta) := \frac{1}{Z(w)} \exp \left( \sum_{n=1}^{N} w_n f_n(\theta) \right) \pi_0(\theta), \)\( \) where \( \pi_1 = \pi \) corresponds to the full density. If \( M \ll N \), evaluating a function proportional to \( \pi_w \) is much less expensive than doing so for the original \( \pi \), resulting in a significant reduction in MCMC computation time. The major challenge posed by this approach, then, is to find a set of weights \( w \) that renders \( \pi_w \) as close as possible to \( \pi \) while maintaining sparsity. Past work [31, 32] formulated this as a sparse regression problem in a Hilbert space with the \( L^2(\hat{\pi}) \) norm for some weighting distribution \( \hat{\pi} \) and vectors \( \hat{g}_n := (f_n - \mathbb{E}_{\hat{\pi}}[f_n]) \),

\[
w^* = \arg \min_{w \in \mathbb{R}^N} \mathbb{E}_{\hat{\pi}} \left[ \left( \sum_{n=1}^{N} \hat{g}_n - \sum_{n=1}^{N} w_n \hat{g}_n \right)^2 \right] \text{ s.t. } w \geq 0, \| w \|_0 \leq M.
\]

As the expectation is generally intractable to compute exactly, a Monte Carlo approximation is used in its place: taking samples \( \{ \theta_i \}_{i=1}^{S} \sim \hat{\pi} \) and setting \( \hat{g}_n = \sqrt{S}^{-1} [g_n(\theta_1) - g_n(\theta_S)] \in \mathbb{R}^S \) where \( g_n = \frac{1}{S} \sum_{s=1}^{S} g_n(\theta_s) \) yields a linear finite-dimensional sparse regression problem in \( \mathbb{R}^S \),

\[
\hat{w}^* = \arg \min_{w \in \mathbb{R}^N} \left\| \sum_{n=1}^{N} \hat{g}_n - \sum_{n=1}^{N} w_n \hat{g}_n \right\|_2^2 \text{ s.t. } w \geq 0, \| w \|_0 \leq M,
\]

which can be solved with sparse optimization techniques [31, 32, 34, 35, 46, 58–60]. However, there are two drawbacks inherent to the Hilbert space formulation. First, the use of the \( L^2(\hat{\pi}) \) norm requires the selection of the weighting function \( \hat{\pi} \), posing a barrier to the full automation of coreset construction. There is currently no guidance on how to select \( \hat{\pi} \), or the effect of different choices in the literature. We show in Sections 4 and 5 that using such a fixed weighting \( \hat{\pi} \) fundamentally limits the quality of coreset construction. Second, the inner products typically cannot be computed exactly, requiring a Monte Carlo approximation. This adds noise to the construction and precludes asymptotic consistency (in the sense that \( \pi_w \not\rightarrow \pi_1 \) as the sparsity budget \( M \rightarrow \infty \)). Addressing these drawbacks is the focus of the present work.

### 3 Bayesian coresets from scratch

In this section, we provide a new formulation of Bayesian coreset construction as variational inference over an exponential family with sparse natural parameters, and develop an iterative greedy algorithm for optimization.

#### 3.1 Sparse exponential family variational inference

We formulate coreset construction as a sparse variational inference problem,

\[
w^* = \arg \min_{w \in \mathbb{R}^N} \text{D}_{\text{KL}}(\pi_w || \pi_1) \text{ s.t. } w \geq 0, \| w \|_0 \leq M.
\]

Expanding the objective and denoting expectations under \( \pi_w \) as \( \mathbb{E}_w \),

\[
\text{D}_{\text{KL}}(\pi_w || \pi_1) = \log Z(1) - \log Z(w) - \sum_{n=1}^{N} (1 - w_n) \mathbb{E}_w [f_n(\theta)] .
\]
Although seemingly ad-hoc, this modification will be placed on a solid information-geometric foundation in Proposition 1 (see also Eq. (34) in Appendix A). Note that since we do not have access to the exact correlations, we must use Monte Carlo estimates via sampling from \( \pi_w \) for greedy selection. Given \( S \) samples \( (\theta_s)_{s=1}^{S} \overset{i.i.d.}{\sim} \pi_w \), these are given by the \( N \)-dimensional vector

\[
\tilde{\text{Corr}} = \text{diag} \left[ \frac{1}{S} \sum_{s=1}^{S} \tilde{g}_s \tilde{g}_s^T \left( \frac{1}{S} \sum_{s=1}^{S} \tilde{g}_s \tilde{g}_s^T \left( 1 - w \right) \right) \right] \tilde{g}_s := \left[ f_{1}(\theta_s) \, \ldots \, f_{N}(\theta_s) \right].
\]

3.2 Greedy selection

One option to build up a coreset incrementally is to use a greedy approach (Algorithm 1) to select and subsequently reweight a single potential function at a time. For greedy selection, the naïve approach is to select the potential that provides the largest local decrease in KL divergence around the current weights \( w \), i.e., selecting the potential with the largest covariance with the residual error per Eq. (9). However, since the weight \( w_n \) will then be optimized over \([0, \infty)\), the selection of the next potential to add should be invariant to scaling each potential \( f_n \) by any positive constant. Thus we propose the use of the correlation—rather than the covariance—between \( f_n \) and the residual error \( f^T (1 - w) \) as the selection criterion:

\[
n^* = \arg \max_{n \in [N]} \{ \text{Corr} \left[ f_n, f^T (1 - w) \right] \} \quad \text{s.t.} \quad w_n > 0, w_n = 0.
\]

Although seemingly ad-hoc, this modification will be placed on a solid information-geometric foundation in Proposition 1 (see also Eq. (34) in Appendix A). Note that since we do not have access to the exact correlations, we must use Monte Carlo estimates via sampling from \( \pi_w \) for greedy selection. Given \( S \) samples \( (\theta_s)_{s=1}^{S} \overset{i.i.d.}{\sim} \pi_w \), these are given by the \( N \)-dimensional vector

\[
\tilde{\text{Corr}} = \text{diag} \left[ \frac{1}{S} \sum_{s=1}^{S} \tilde{g}_s \tilde{g}_s^T \left( \frac{1}{S} \sum_{s=1}^{S} \tilde{g}_s \tilde{g}_s^T \left( 1 - w \right) \right) \right] \tilde{g}_s := \left[ f_{1}(\theta_s) \, \ldots \, f_{N}(\theta_s) \right].
\]
Algorithm 1. As in the greedy selection step, the cost of each gradient step is within these indices. Note that this computation has cost $O(NS)$. If $N$ is large enough that computing the entire vectors $\hat{g}_s \in \mathbb{R}^N$ is cost-prohibitive, one may instead compute $\hat{g}_s$ in Eq. (11) only for indices in $I \cup U$—where $I = \{ n \in [N] : w_n > 0 \}$ is the set of active indices, and $U$ is a uniformly selected subsample of $U \in [N]$ indices—and perform greedy selection only within these indices.

3.3 Weight update

After selecting a new potential function $n^*$, we add it to the active set of indices $I \subseteq [N]$ and update the weights by optimizing

$$w^* = \arg \min_{w \in \mathbb{R}^N} D_{KL}(\pi_w || \pi) \quad \text{s.t.} \quad v \geq 0, \quad (1 - 1_I)^T v = 0.$$ (12)

In particular, we run $T$ steps of generating $S$ samples $(\theta_s)_{s=1}^S \overset{iid}{\sim} \pi_w$, computing a Monte Carlo estimate $D$ of the gradient $\nabla_w D_{KL}(\pi_w || \pi_1)$ based on Eq. (3),

$$D := -\frac{1}{S} \sum_{s=1}^S \hat{g}_s \hat{g}_s^T (1 - w) \in \mathbb{R}^N \quad \hat{g}_s \text{ as in Eq. (11)},$$ (13)

and taking a stochastic gradient step $w_n \leftarrow w_n - \gamma_t D_n$ at step $t \in [T]$ for each $n \in I$, using a typical learning rate $\gamma_t \propto t^{-1}$. The details of the weight update step are shown in lines 10–15 of Algorithm 1. As in the greedy selection step, the cost of each gradient step is $O(NS)$, due to the $\hat{g}_s^T 1$ term in the gradient. If $N$ is large enough that this computation is cost-prohibitive, one can use $\hat{g}_s$ computed only for indices in $I \cup U$, where $U$ is a uniformly selected subsample of $U \in [N]$ indices.

4 The information geometry of coreset construction

The perspective of coresets as a sparse exponential family also enables the use of information geometry to derive a unifying connection between the variational formulation and previous constructions. In particular, the family of coreset posteriors defines a Riemannian statistical manifold $\mathcal{M} = \{ \pi_w \}_{w \in \mathbb{R}_{\geq 0}^N}$, with chart $\mathcal{M} \rightarrow \mathbb{R}_{\geq 0}^N$, endowed with the Fisher information metric $G$ [61, p. 33,34],

$$G(w) = \int \pi_w(\theta) \nabla \log \pi_w(\theta) \nabla \log \pi_w(\theta)^T d\theta = \nabla^2 \log Z(w) = \text{Cov}_w [f].$$ (14)

For any differentiable curve $\gamma : [0, 1] \rightarrow \mathbb{R}_{\geq 0}^N$, the metric defines a notion of path length,

$$L(\gamma) = \int_0^1 \sqrt{\frac{d\gamma(t)}{dt}^T G(\gamma(t)) \frac{d\gamma(t)}{dt}} dt,$$ (15)

and a constant-speed curve of minimal length between any two points $w, w' \in \mathbb{R}_{\geq 0}^N$ is referred to as a geodesic [61] Thm. 5.2. The geodesics are the generalization of straight lines in Euclidean space to curved Riemannian manifolds, such as $\mathcal{M}$. Using this information-geometric view, Proposition 1 shows that both Hilbert coreset construction (Eq. (3)) and the proposed greedy sparse variational inference procedure (Algorithm 1) attempt to directionally align the $\hat{w} \rightarrow w$ and $\hat{w} \rightarrow 1$ geodesics on $\mathcal{M}$ for $\hat{w}, w, 1 \in \mathbb{R}_{\geq 0}^N$ (reference, coreset, and true posterior weights, respectively) as illustrated in Fig. 1. The key difference is that Hilbert coreset construction uses a fixed reference point $\hat{w}$—corresponding to $\hat{\pi}$ in Eq. (3)—and thus operates entirely in a single tangent space of $\mathcal{M}$, while the proposed greedy method uses $\hat{w} = w$ and thus improves its tangent space approximation as the algorithm iterates. For this reason, we refer to the method in Section 3 as a Riemannian coreset construction algorithm. In addition to this unification of coreset construction methods, the geometric perspective also provides the means to show that the Hilbert coresets objective bounds the symmetrized coreset KL divergence $D_{KL}(\pi_w || \pi) + D_{KL}(\pi || \pi_w)$ if the Riemannian metric does not vary too much, as shown in Proposition 2. Incidentally, Lemma 3 in Appendix A—which is used to prove Proposition 2—also provides a nonnegative unbiased estimate of the symmetrized coreset KL divergence, which may be used for performance monitoring in practice.
Algorithm 1: Greedy sparse stochastic variational inference

1: procedure SPARSEVI(f, π₀, S, (γᵢ)ᵢ=1, M)
2:   w ← 0 ∈ ℝᴺ, ℐ ← ∅
3: for m = 1, ..., M do
4:   (θₛ)ₛ=1的功能 in Eq. (3) is the objective function in Eq. (3). In particular, for GIGA we used a 100-dimensional random projection generated from a Gaussian (32). In particular, if GIGA we used a 100-dimensional random projection generated from a Gaussian (32). In particular, if GIGA we used a 100-dimensional random projection generated from a Gaussian (32). In particular, if GIGA we used a 100-dimensional random projection generated from a Gaussian (32).

5:   Compute the N-dimensional potential vector for each sample
6:   ſₛ ← f(θₛ) ∈ ℝᴺ for s ∈ [S], and ſ ← 1/N ∑ₛ=1 ſₛ
7:   ̂gₛ ← ſₛ − ſ for s ∈ [S]
8:   Estimate correlations between the potentials and the residual error
9:   Corr ← diag [1/N ∑ₛ=1 ̂gₛ ̂gₛᵀ] − [1/N ∑ₛ=1 ̂gₛ ̂gₛᵀ (1 − w)] ∈ ℝᴺ
10:  Add the best next potential to the coreset
11:  n* ← arg maxₙ∈[N] [Corrₙ] [n ∈ ℐ] + [Corrₙ] [n ∉ ℐ]
12:  ℐ ← ℐ ∪ {n*}
13:  Update all the active weights in ℐ via stochastic gradient descent on DKL (π₁ || π)
14: end for
15: return w
16: end procedure

Proposition 1. Suppose ̂π in Eq. (3) satisfies ̂π = π₀ for a set of weights ̂w ∈ ℝᴺ₀. For u, v ∈ ℝᴺ₀, let ̂ξₜ→ₚ denote the initial tangent of the u → v geodesic on M, and ⟨·,·⟩₀ denote the inner product under the Riemannian metric G(u) with induced norm ∥·∥₀. Then Hilbert coreset construction in Eq. (3) is equivalent to

\[ w* = \arg \min_{w ∈ ℝᴺ} ||ξₜ→ₚ − ξₜ→ₚ ||₀ \quad s.t. \quad w ≥ 0, \quad ||w||₀ ≤ M, \quad (16) \]

and each greedy selection step of Riemannian coreset construction in Eq. (16) is equivalent to

\[ n* = \arg \min_{n ∈ [N], t₂ ∈ ℝ} ||ξₜ→ₚ − ξₜ→ₚ + t₂₁ₙ ||₀ \quad s.t. \quad n /∈ ℐ, \quad t₂ > 0. \quad (17) \]

Proposition 2. Suppose ̂π in Eq. (3) satisfies ̂π = π₀ for a set of weights ̂w ∈ ℝᴺ₀. Then if Jₚ(̂w) is the objective function in Eq. (3),

\[ D_{KL}(π || π₀) + D_{KL}(π₀ || π) ≤ Cₚ(π) · Jₚ(π), \quad (18) \]

where Cₚ(π) := Eᵤ∼Unif[0,1] [λₘₐₓ(G(u)⁻¹/2 G((1 − U)u + U₁)G(u)⁻¹/2)]. In particular, if ∇ₚ² log Z(π) is constant in w ∈ ℝᴺ₀, then Cₚ(π) = 1.

5 Experiments

In this section, we compare the quality of coresets constructed via the proposed SPARSEVI greedy coreset construction method, uniform random subsampling, and Hilbert coreset construction (GIGA [32]). In particular, for GIGA we used a 100-dimensional random projection generated from a Gaussian π with two parametrizations: one with mean and covariance set using the moments of the exact posterior (Optimal) which is a benchmark but is not possible to achieve in practice; and one
Figure 1: Information-geometric view of greedy coreset construction on the coreset manifold $\mathcal{M}$. (1a): Hilbert coreset construction, with weighting distribution $\pi^{\hat{w}}$, full posterior $\pi$, coreset posterior $\pi^{w}$, and arrows denoting initial geodesic directions from $\hat{w}$ towards new datapoints. (1b): Riemannian coreset construction, with the path of posterior approximations $\pi^{w}_t$, $t = 0, \ldots, 3$, and arrows denoting initial geodesic directions towards new datapoints to add within each tangent plane.

Figure 2: (2a): Synthetic comparison of coreset construction methods. Solid lines show the median KL divergence over 10 trials, with 25th and 75th percentiles shown by shaded areas. (2b): 2D projection of coresets after 0, 1, 5, 20, 50, and 100 iterations via SparseVI. True/coreset posterior and $2\sigma$-predictive ellipses are shown in black/blue respectively. Coreset points are black with radius denoting weight.

with mean and covariance uniformly distributed between the prior and the posterior with 75% relative noise added (Realistic) to simulate the choice of $\hat{\pi}$ without exact posterior information. Experiments were performed on a machine with an Intel i7 8700K processor and 32GB memory; code is available at [www.github.com/trevorcampbell/bayesian-coresets](http://www.github.com/trevorcampbell/bayesian-coresets).

5.1 Synthetic Gaussian posterior inference

We first compared the coreset construction algorithms on a synthetic example involving posterior inference for the mean of a $d$-dimensional Gaussian with Gaussian observations,

$$\theta \sim \mathcal{N}(\mu_0, \Sigma_0) \quad x_n \overset{i.i.d.}{\sim} \mathcal{N}(\theta, \Sigma), \quad n = 1, \ldots, N. \quad (19)$$

We selected this example because it decouples the evaluation of the coreset construction methods from the concerns of stochastic optimization and approximate posterior inference: the coreset posterior $\pi^w$ is a Gaussian $\pi^w = \mathcal{N}(\mu_w, \Sigma_w)$ with closed-form expressions for the parameters as well as covariance (see Appendix [B] for the derivation),

$$\Sigma_w = (\Sigma_0^{-1} + \sum_{n=1}^N w_n \Sigma^{-1})^{-1} \quad \Sigma_w = \Sigma_w (\Sigma_0^{-1} \mu_0 + \Sigma^{-1} \sum_{n=1}^N w_n x_n) \quad (20)$$

$$\text{Cov}_w[f_n, f_m] = \frac{1}{2} \text{tr} \Psi^T \Psi + \nu_m^T \Psi \nu_n, \quad (21)$$
We generated 50 basis functions for each of 6 scales $\nu$ and $T = 100$ via SparseVI compared with the exact posterior. Posterior and final coreset highlighted in the top row. Coreset points are black with radius denoting weight.

where $\Sigma = QQ^T$, $\nu_n := Q^{-1}(x_n - \mu_w)$ and $\Psi := Q^{-1}\Sigma w Q^{-T}$. Thus the greedy selection and weight update can be performed without Monte Carlo estimation. We set $\Sigma_0 = \Sigma = I$, $\mu_0 = 0$, $d = 200$, and $N = 1,000$. We used a learning rate of $\gamma_t = t^{-1}$, $T = 100$ weight update optimization iterations, and $M = 200$ greedy iterations, although note that this is an upper bound on the size of the coreset as the same data point may be selected multiple times. The results in Fig. 2 demonstrate that the use of a fixed weighting function $\pi$ (and thus, a fixed tangent plane on the coreset manifold) fundamentally limits the quality coresets via past algorithms. In contrast, the proposed greedy algorithm is “manifold-aware” and is able to continually improve the approximation, resulting in orders-of-magnitude improvements in KL divergence to the true posterior.

5.2 Bayesian radial basis function regression

Next, we compared the coreset construction algorithms on Bayesian basis function regression for $N = 10,000$ records of house sale log-price $y_n \in \mathbb{R}$ as a function of latitude / longitude coordinates $x_n \in \mathbb{R}^2$ in the UK.

The regression problem involved inference for the coefficients $\alpha \in \mathbb{R}^K$ in a linear combination of radial basis functions $b_k(x) = \exp(-\frac{1}{2}\sigma_k^2(x - \mu_k)^2)$, $k = 1, \ldots, K$,

$$y_n = b_n^T \alpha + \epsilon_n \quad \epsilon_n \sim \mathcal{N}(0, \sigma^2) \quad b_n = [b_1(x_n) \ldots b_K(x_n)]^T \quad \alpha \sim \mathcal{N}(\mu_0, \sigma_0^2 I). \quad (22)$$

We generated 50 basis functions for each of 6 scales $\sigma_k \in \{0.2, 0.4, 0.8, 1.2, 1.6, 2.0\}$ by generating means $\mu_k$ uniformly from the data, and added one additional near-constant basis with scale 100 and mean corresponding to the mean latitude and longitude of the data. This resulted in $K = 301$ total basis functions and thus a 301-dimensional regression problem. We set the prior and noise parameters $\mu_0, \sigma_0^2, \sigma^2$ equal to the empirical mean, second moment, and variance of the price paid $(y_n)_{n=1}^N$ across the whole dataset, respectively. As in Section 5.1 the posterior and log-likelihood covariances are available in closed form, and all algorithmic steps can be performed without Monte Carlo. In particular, $\pi_w = \mathcal{N}(\mu_w, \Sigma_w)$, where (see Appendix B for the derivation)

$$\Sigma_w = (\Sigma_0^{-1} + \sigma^{-2} \sum_{n=1}^N w_n b_n b_n^T)^{-1} \quad \text{and} \quad \mu_w = \Sigma_w (\Sigma_0^{-1} \mu_0 + \sigma^{-2} \sum_{n=1}^N w_n y_n b_n) \quad (23)$$

$$\text{Cov}_w [\ell_n, \ell_m] = \sigma^{-4} (\nu_n \nu_m \beta_n^T \beta_m + 1/2 (\beta_n^T \beta_m)^2). \quad (24)$$

where $\nu_n := y_n - \mu_w^T b_n, \Sigma_w = LL^T$, and $\beta_n := L^T b_n$. We used a learning rate of $\gamma_t = t^{-1}$, $T = 100$ optimization steps, and $M = 300$ greedy iterations, although again note that this is an upper

---

Figure 4: The results of the logistic (4a) and Poisson (4b) regression experiments. Plots show the median KL divergence (estimated using the Laplace approximation [62] and normalized by the value for the prior) across 10 trials, with 25th and 75th percentiles shown by shaded areas. From top to bottom, (4a) shows the results for logistic regression on synthetic, chemical reactivities, and phishing websites data, while (4b) shows the results for Poisson regression on synthetic, bike trips, and airport delays data. See Appendix C for details.

bound on the coreset size. The results in Fig. 3 generally align with those from the previous synthetic experiment. The proposed sparse variational inference formulation builds coresets of comparable quality to Hilbert coreset construction (when given the exact posterior for $\hat{\pi}$) up to a size of about 150. Beyond this point, past methods become limited by their fixed tangent plane approximation while the proposed method continues to improve. This experiment also highlights the sensitivity of past methods to the choice of $\hat{\pi}$: uniform subsampling outperforms GIGA with a realistic choice of $\hat{\pi}$.

5.3 Bayesian logistic and Poisson regression

Finally, we compared the methods on logistic and Poisson regression applied to six datasets (details may be found in Appendix C) with $N = 500$ and dimension ranging from 2-15. We used $M = 100$ greedy iterations, $S = 100$ samples for Monte Carlo covariance estimation, and $T = 500$ optimization iterations with learning rate $\gamma_t = 0.5t^{-1}$. Fig. 4 shows the result of this test, demonstrating that the proposed greedy sparse VI method successfully recovers a coreset with divergence from the exact posterior as low or lower than GIGA with without having the benefit of a user-specified weighting function. Note that there is a computational price to pay for this level of automation; Fig. 5 Appendix C shows that $\text{SparseVI}$ is significantly slower than Hilbert coreset construction via GIGA [32], primarily due to the expensive gradient descent weight update. However, if we remove GIGA (Optimal) from consideration due to its unrealistic use of $\hat{\pi} \approx \pi_1$, $\text{SparseVI}$ is the only practical coreset construction algorithm that reduces the KL divergence to the posterior appreciably for reasonable coreset sizes. We leave improvements to computational cost for future work.

6 Conclusion

This paper introduced sparse variational inference for Bayesian coreset construction. By exploiting the fact that coreset posteriors form an exponential family, a greedy algorithm as well as a unifying Riemannian information-geometric view of present and past coreset constructions were developed. Future work includes extending sparse VI to improved optimization techniques beyond greedy methods, and reducing computational cost.

Acknowledgments T. Campbell and B. Beronov are supported by National Sciences and Engineering Research Council of Canada (NSERC) Discovery Grants. T. Campbell is additionally supported by an NSERC Discovery Launch Supplement.
References


