Bayesian Coresets: Revisiting the Nonconvex Optimization Perspective

Jacky Y. Zhang[1] yiboz@illinois.edu
Rajiv Khanna[2] rajivak@berkeley.edu
Anastasios Kyrillidis[3] anastasios@rice.edu
Oluwasanmi Koyejo[1] sanmi@illinois.edu
[1]University of Illinois at Urbana-Champaign [2]University of California, Berkeley [3]Rice University

Abstract

Bayesian coresets have emerged as a promising approach for implementing scalable Bayesian inference. The Bayesian coreset problem involves selecting a (weighted) subset of the data samples, such that the posterior inference using the selected subset closely approximates the posterior inference using the full dataset. This manuscript revisits Bayesian coresets through the lens of sparsity constrained optimization. Leveraging recent advances in accelerated optimization methods, we propose and analyze a novel algorithm for coreset selection. We provide explicit convergence rate guarantees and present an empirical evaluation on a variety of benchmark datasets to highlight our proposed algorithm's superior performance compared to state-of-the-art on speed and accuracy.

1 Introduction

Bayesian coresets have emerged as a promising approach for scalable Bayesian inference (Huggins et al., 2016; Campbell & Broderick, 2018, 2019; Campbell & Beronov, 2019). The key idea is to select a (weighted) subset of the data such that the posterior inference using the selected subset closely approximates the posterior inference using the full dataset. This creates a trade-off, where using Bayesian coresets as opposed to the full dataset exchanges approximation accuracy for computational speedups. We study Bayesian coresets as they are easy to implement, effective in practice, and come with useful theoretical guarantees that relate the coreset size with the approximation quality.

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

The main technical challenge in the Bayesian coreset problem lies in handling the combinatorial constraints – we desire to select a few data points out of many as the coreset. In terms of optimization, previous approaches mainly rely on two ideas: *convexification* and *greedy methods*. In convexification (Campbell & Broderick, 2019), the sparsity constraint – i.e., selection of k data samples – is relaxed into a convex ℓ_1-norm constraint. This allows them to use out-of-the-box solvers such as Frank-Wolfe (FW) type-of methods (Frank & Wolfe, 1956; Jaggi, 2013). An alternative approach is by using greedy methods (Campbell & Broderick, 2018), which constructs a sparse weight vector based on local decisions to greedily optimize the approximation problem (Tropp & Gilbert, 2007; Needell & Tropp, 2009). The resulting method, greedy iterative geodesic ascent (GIGA), achieves linear convergence with no hyper-parameter tuning and optimal scaling (Campbell & Broderick, 2018). More recently, sparse variational inference (SparseVI) is considered for Bayesian coreset construction. SparseVI also employs a greedy algorithm to minimize a KL divergence objective. The method achieves state-of-the-art accuracy, but at a cost of higher computational requirements. Therefore, existing work illustrates the trade-off between accuracy and efficiency, opening a gap for improvements.

We revisit Bayesian coresets through the lens of sparsity constrained optimization. Sparsity, a kind of nonconvexity, appears in a variety of applications in machine learning and statistics. For instance, compressed sensing (Donoho et al., 2006; Candes, 2008) is an example where sparsity is used as a complexity measure for signal representation. Leveraging and building upon recent advances in non-convex optimization, we solve the Bayesian coreset problem based on hard thresholding algorithms (Blumensath & Davies, 2009) that directly work on the non-convex sparsity constraint. Hard-thresholding schemes are highly flexible, and easily accommodate variations such as subspace exploration (Dai & Milenkovic, 2009), de-bias steps (Needell & Tropp, 2009), adaptive step size selections (Kyrillidis & Cevher, 2011), as well as different types of spar-

sity constraints, such as group sparsity (Baldassarre et al., 2016), sparsity within groups (Kyrillidis et al., 2015), and generic structured sparsity (Baraniuk et al., 2010). The thresholding step involves a projection onto the k-sparsity constraint set to determine the selected sample set in each iteration. While we achieve state-of-the-art accuracy using direct application of this algorithm, re-building the set in every iteration makes it slower than previous works. To fix this, we employ line search for step size selection and momentum based techniques (Khanna & Kyrillidis, 2018) to accelerate the algorithm, also achieving state-of-the-art speed.

Contributions. In this paper, we adapt accelerated iterative hard thresholding schemes to the Bayesian coreset problem. Despite directly attacking the nonconvex optimization problem, we provide strong convergence guarantees. To summarize our contributions:

- We revisit the Bayesian coreset problem via a nonconvex (sparse) optimization lens, and provide an IHT-based algorithm that combines hard thresholding and momentum steps;
- We analyze its convergence based on standard assumptions;
- We provide extensive empirical evaluation[1] to show superior performance of the proposed method vis-à-vis state-of-the-art algorithms in terms of approximation accuracy as well as speed.

2 Problem Formulation

Given n observations, one can compute the log-likelihood $\mathcal{L}_i(\theta)$ of each of the observations, parameterized by θ. Assuming observations are conditionally independent given θ, one can represent the likelihood of all the observations as the sum of individual log-likelihoods, i.e., $\mathcal{L}(\theta) = \sum_{i=1}^n \mathcal{L}_i(\theta)$. With prior density $\pi_0(\theta)$, the posterior density can be derived as:

$$\pi(\theta) := \tfrac{1}{Z} \cdot e^{\mathcal{L}(\theta)} \cdot \pi_0(\theta),$$

where $Z = \int e^{\mathcal{L}(\theta)} \pi_0(\theta) d\theta$ is a normalization factor.

However, for most applications, exact posterior estimation is intractable; *i.e.*, π is too hard to evaluate exactly. Practitioners use algorithms for approximate inference that may approximate the π in a closed-form (*e.g.*, using variational inference), or allow for sampling from the posterior without providing a closed-form expression (*e.g.*, MCMC methods). Such algorithms often scale at least linearly with the size of the dataset n, which makes them prohibitively expensive for large datasets. As such, designing algorithms to speed up inference is an area of active research.

One solution to the scalability problem is to use coresets. Coresets approximate the empirical log-likelihood $\mathcal{L} = \sum_{i=1}^n \mathcal{L}_i$ using a *weighted sum of a subset of all the log-likelihoods* $\mathcal{L}_i$. In other words, we use $\mathcal{L}_w = \sum_{i=1}^n w_i \mathcal{L}_i$ to approximate the true $\mathcal{L}$, where $w \in \mathbb{R}_+^n$ is a non-negative sparse vector. It will be useful to view that $\mathcal{L}, \mathcal{L}_i$ and $\mathcal{L}_w$ are functions in a Hilbert space, and we will use L^2-norm to denote the 2-norm defined in function space, differentiating with the ℓ_2-norm defined in Euclidean space. We enforce the sparsity constraint as $\|w\|_0 \leq k$, for $k < n$; here $\|\cdot\|_0$ denotes the pseudo-norm that counts the number of non-zero entries.

When $k < n$, posterior estimation (e.g., using MCMC or variational inference) is less expensive on the coreset as opposed to the entire dataset. However, sparsifying w involves dropping some samples, which in turn implies deviating from the best performance possible from using the full dataset. The Bayesian coreset problem is formulated to minimize this loss in performance.

The Bayesian Coreset Problem. *The Bayesian coreset problem is to control the deviation of coreset log-likelihood from true log-likelihood via sparsity:*

$$\begin{aligned} \underset{w \in \mathbb{R}^n}{\arg\min} \quad & f(w) := \textsc{Dist}(\mathcal{L},\ \mathcal{L}_w) \\ \text{s.t.} \quad & \|w\|_0 \leq k,\ w_i \geq 0, \forall i. \end{aligned} \tag{1}$$

Key components are (i) the weights $w \in \mathbb{R}_+^n$ over n data points, (ii) the function $f(\cdot)$ that controls the deviation between the full-dataset log-likelihood $\mathcal{L}$ and the coreset log-likelihood $\mathcal{L}_w$ using the distance functional $\textsc{Dist}(\cdot,\ \cdot)$, and (iii) the non-convex sparsity constraint that restricts the number of nonzeros in w, thus constraining the number of active data points in the coreset. Examples of $\textsc{Dist}(\cdot,\ \cdot)$ include the weighted L^2-norm (Campbell & Broderick, 2019) and the KL-divergence (Campbell & Beronov, 2019). In this manuscript, we consider the $L^2(\hat{\pi})$-norm as the distance metric in the embedding Hilbert space, *i.e.*,

$$\begin{aligned} \textsc{Dist}(\mathcal{L},\ \mathcal{L}_w)^2 &= \|\mathcal{L} - \mathcal{L}_w\|_{\hat{\pi},2}^2 \\ &= \mathbb{E}_{\theta \sim \hat{\pi}} \left[(\mathcal{L}(\theta) - \mathcal{L}_w(\theta))^2 \right], \end{aligned} \tag{2}$$

where $\hat{\pi}$ is a weighting distribution that has the same support as true posterior π. Ideally, $\hat{\pi}$ is the true posterior, which is obviously unknown. However, one can employ Laplace approximation to derive an inexpensive and reasonable approximation for $\hat{\pi}$ (Campbell & Broderick, 2019).

To account for the shift invariance, we write $g_i = \mathcal{L}_i - \mathbb{E}_{\theta \sim \hat{\pi}} \mathcal{L}_i(\theta)$, so the equivalent optimization problem is now: minimize $\| \sum_{i=1}^n g_i - \sum_{i=1}^n w_i g_i \|_{\hat{\pi},2}^2$. Further,

[1] Code available at `https://github.com/jackyzyb/bayesian-coresets-optimization`

noting that the $L^2(\hat{\pi})$-norm is in the form of expectation (equation (2)), it can be approximated by a finite-dimensional ℓ_2-norm which replaces the function with a vector of sampled evaluations $\theta \sim \hat{\pi}$, *i.e.*, its Monte Carlo approximation. Thus, given S samples $\{\theta_j\}_{j=1}^S, \theta_j \sim \hat{\pi}$, and using

$$\hat{g}_i = \tfrac{1}{\sqrt{S}} \cdot \left[\mathcal{L}_i(\theta_1) - \bar{\mathcal{L}}_i, \ldots, \mathcal{L}_i(\theta_S) - \bar{\mathcal{L}}_i\right]^\top \in \mathbb{R}^S$$

as projections from function space to standard Euclidean space, where $\bar{\mathcal{L}}_i = \frac{1}{S}\sum_{j=1}^S \mathcal{L}_i(\theta_j)$, the Bayesian coreset problem (1) becomes a *finite-dimensional sparse regression problem:*

$$\begin{aligned} \underset{w\in\mathbb{R}^n}{\arg\min} \quad & f(w) := \left\| \sum_{i=1}^n \hat{g}_i - \sum_{i=1}^n w_i \hat{g}_i \right\|_2^2 \\ \text{s.t.} \quad & \|w\|_0 \le k, \quad w_i \ge 0, \forall i. \end{aligned} \tag{3}$$

The resulting sparse regression problem is non-convex due to the combinatorial nature of the constraints. Previous methods that use this ℓ_2-norm formulation (Campbell & Broderick, 2019, 2018) offers less satisfactory approximation accuracy compared to the state-of-the-art sparse variational inference method (Campbell & Beronov, 2019). However, the high computational cost of the latter method makes it impractical for real-world large datasets. Nonetheless, as we will show, our approach for solving equation (3) using a variant of iterative hard thresholding, achieves better accuracy and speed.

3 Our approach

Algorithm 1 Vanilla IHT

input Objective $f : \mathbb{R}^n \to \mathbb{R}$; sparsity k; step size μ

1: Initialize w
2: **repeat**
3: $\quad w \leftarrow \Pi_{\mathcal{C}_k \cap \mathbb{R}^n_+}\left(w - \mu \nabla f(w)\right)$
4: **until** Stop criteria met
5: **return** w

For clarity of exposition, we gradually build up our approach for solving the optimization problem (3). The fundamental ingredient of our approach is the vanilla Iterative Hard Thresholding (IHT) method presented in Algorithm 1. We develop our approach by augmenting IHT with momentum updates, step size selection for line search and active subspace expansion techniques to accelerate and automate the algorithm (Algorithms 2 & 3). Details follow.

3.1 Iterative Hard Thresholding (IHT)

The classical IHT (Blumensath & Davies, 2009) is a projected gradient descent method that performs a gradient descent step and then projects the iterate onto the non-convex k-sparsity constraint set. We denote the orthogonal projection of a given $z \in \mathbb{R}^n$ to a space $\mathcal{C} \subseteq \mathbb{R}^n$ as: $\Pi_{\mathcal{C}}(z) := \arg\min_{w\in\mathcal{C}} \|w - z\|_2$. Define the sparsity restricted space as: $\mathcal{C}_k = \left\{w \in \mathbb{R}^n : |\text{supp}(w)| \le k\right\}$, where $\text{supp}(w) = \{i | w_i \neq 0\}$ denotes the support set of w. Here, we describe the plain sparsity case, but one can consider different realizations of $\mathcal{C}_k$ as in (Baldassarre et al., 2016; Kyrillidis et al., 2015; Baraniuk et al., 2010). The projection step in the classical IHT, *i.e.,*, $\Pi_{\mathcal{C}_k}$, can be computed easily by selecting the top-k elements in $O(n \log k)$ time; but projection can be more challenging for more complex constraint sets, e.g., if the variable is a distribution on a lattice (Zhang et al., 2019).

For our problem, we require that the projected sparse vector only has non-negative values. For vector variate functions, the projection step in Algorithm 1, i.e., $\Pi_{\mathcal{C}_k \cap \mathbb{R}^n_+}(w)$ is also straightforward; it can be done optimally in $O(n \log k)$ time by simply picking the top k largest *non-negative* elements. More discussions about the projections are presented in section B in appendix.

3.2 Accelerated IHT

For clarity, we rewrite the problem in equation (3) as:

$$w^* = \underset{w\in\mathcal{C}_k\cap\mathbb{R}^n_+}{\arg\min} \quad f(w) := \|y - \Phi w\|_2^2,$$

where $y = \sum_{i=1}^n \hat{g}_i$ and $\Phi = [\hat{g}_1, \ldots, \hat{g}_n]$. In this case, $\nabla f(w) \equiv -2\Phi^\top(y - \Phi w)$.

Step size selection in IHT: Classical results on the performance of IHT algorithms come with rigorous convergence guarantees (under regularity conditions) (Blumensath & Davies, 2009; Foucart, 2011). However, these results require step size assumptions that either do not work in practice, or rely on strong assumptions. For example, in (Blumensath & Davies, 2009; Foucart, 2011) strong isometry constant bounds are assumed to allow step size $\mu = 1$ for all the iterations, and thus remove the requirement of hyper-parameter tuning. Moreover, the authors in (Blumensath & Davies, 2010) present toy examples by carefully selecting Φ so that the vanilla IHT algorithm diverges without appropriate step size selection. In this work, given the quadratic objective $f(w)$, we perform exact line search to obtain the best step size per iteration (Blumensath & Davies, 2010; Kyrillidis & Cevher, 2011): $\mu_t := \|\widetilde{\nabla}_t\|_2^2 / 2\|\Phi\widetilde{\nabla}_t\|_2^2$; details in Algorithm 2.

Memory in vanilla IHT: Based upon the same ideas as step size selection, we propose to include adaptive momentum acceleration; we select the momentum term as the minimizer of the objective: $\tau_{t+1} = \arg\min_\tau f(w_{t+1} + \tau(w_{t+1} - w_t)) =$

Algorithm 2 Automated Accelerated IHT (A-IHT)

input Objective $f(w) = \|y - \Phi w\|_2^2$; sparsity k

1: $t = 0$, $z_0 = 0$, $w_0 = 0$
2: **repeat**
3: $\quad \mathcal{Z} = \text{supp}(z_t)$
4: $\quad \mathcal{S} = \text{supp}(\Pi_{\mathcal{C}_k \setminus \mathcal{Z}}(\nabla f(z_t))) \cup \mathcal{Z}$ where $|\mathcal{S}| \leq 3k$
5: $\quad \widetilde{\nabla}_t = \nabla f(z_t)\big|_{\mathcal{S}}$
6: $\quad \mu_t = \arg\min_\mu f(z_t - \mu \widetilde{\nabla}_t) = \frac{\|\widetilde{\nabla}_t\|_2^2}{2\|\Phi \widetilde{\nabla}_t\|_2^2}$
7: $\quad w_{t+1} = \Pi_{\mathcal{C}_k \cap \mathbb{R}_+^n}(z_t - \mu_t \nabla f(z_t))$
8: $\quad \tau_{t+1} = \arg\min_\tau f(w_{t+1} + \tau(w_{t+1} - w_t))$
$\qquad = \frac{\langle y - \Phi w_{t+1}, \Phi(w_{t+1} - w_t)\rangle}{2\|\Phi(w_{t+1} - w_t)\|_2^2}$
9: $\quad z_{t+1} = w_{t+1} + \tau_{t+1}(w_{t+1} - w_t)$
10: $\quad t = t + 1$
11: **until** Stop criteria met
12: **return** w_t

$\frac{\langle y - \Phi w_{t+1}, \Phi(w_{t+1} - w_t)\rangle}{2\|\Phi(w_{t+1} - w_t)\|_2^2}$, which also comes out as a closed-form solution. The step $z_{t+1} = w_{t+1} + \tau_{t+1}(w_{t+1} - w_t)$ at the end of the algorithm captures memory in the algorithm based on the results on acceleration by Nesterov (1983) for convex optimization.

Automated Accelerated IHT for coreset selection: Combining the ideas above leads to Automated Accelerated IHT, as presented in Algorithm 2. The algorithm alternates between the projection step (steps 6 and 7) after the gradient updates, and the momentum acceleration step (step 8). It thus maintains two sets of iterates that alternatively update each other in each iteration at only a constant factor increase in per iteration complexity. The iterate w_t at iteration t is the most recent estimate of the optimizer, while the iterate z_t models the effect of momentum or "memory" in the iterates. We have shown exact line search that solves one dimensional problems to automate the step size selection (μ) and the momentum parameter (τ) for acceleration. In practice, these parameters can also be selected using a backtracking line search.

Using de-bias steps in Automated Accelerated IHT: Based on pursuit methods for sparse optimization (Needell & Tropp, 2009; Dai & Milenkovic, 2009; Kyrillidis & Cevher, 2014), we propose a modification that improves upon Algorithm 2 both in speed and accuracy in empirical evaluation. The modified algorithm is presented in Algorithm 3 in section A in appendix due to space limitations. The key differences of Algorithm 3 from Algorithm 2 are that, with additional de-bias steps, one performs another gradient step and a line search in the sparsified space in each iteration for further error reduction. We omit these steps in the algorithmic description to maintain clarity, since these steps do not provide much intellectual merit to the existing algorithm, but help boost the practical performance of Automated Accelerated IHT.

Time complexity analysis. Here, we analyze the time complexity of IHT in terms of the dataset size n and coreset size k, and show that IHT is faster than previous methods for Bayesian coreset construction. We take Algorithm 2 as an example and let the stopping criteria be a constant constraint on number of iterations; the time complexity for all the three versions of IHT (*i.e.*, Algorithm 1, 2, 3) are the same. As the dimension of z_t, w_t is n, and the matrix multiplication Φw has complexity $O(n)$, we can see that each line in Algorithm 2 except for the projection steps (line 4 and line 7) have complexity $O(n)$. The projection steps, as we have discussed in subsection 3.1, can be done in $O(n \log k)$. Therefore, the total time complexity of IHT is $O(n \log k)$. In comparison, previous state-of-the-art algorithms GIGA (Campbell & Broderick, 2018) and SparseVI (Campbell & Beronov, 2019) have time complexity $O(nk)$, which is exponentially slower than IHT in terms of coreset size k. We note that some other factors play a role in the time complexity, *e.g.*, the number of samples from posterior for IHT, GIGA and SparseVI; the number of iterations of the stochastic gradient descent in SparseVI. However, unlike n and k defined by the problem, those factors are chosen parameters specific to each algorithm. Therefore, we treat them as pre-specified constants, and focus on the complexity *w.r.t.* dataset size n and coreset size k.

3.3 Theoretical Analysis of Convergence

In this subsection, we study the convergence properties of our main algorithm Automated Accelerated IHT in Algorithm 2. We make a standard assumption about the objective – the Restricted Isometry Property or RIP (Assumption 1), which is a standard assumption made for analysis of IHT and its variants.

Assumption 1 (Restricted Isometry Property (RIP))**.** *The matrix Φ in the objective function satisfies the RIP property,* i.e.*, for $\forall w \in \mathcal{C}_k$*

$$\alpha_k \|w\|_2^2 \leq \|\Phi w\|_2^2 \leq \beta_k \|w\|_2^2.$$

In RIP, α_k reflects the convexity and β_k reflects the smoothness of the objective in some sense (Khanna & Kyrillidis, 2018; Kyrillidis & Cevher, 2014). We note that the assumption may not be necessary but is sufficient to show convergence theoretically. For example, if the number of samples required to exactly construct $\hat{g}$ is less than the coreset size ($a_k = 0$ in RIP), so that the system becomes under-determined, then a local minimum can also be global achieving zero error without assuming that the RIP holds. On the other hand, when the number of samples goes to infinity, RIP is saying that the restricted eigenvalues of

covariance matrix, $cov[\mathcal{L}_i(\theta), \mathcal{L}_j(\theta)]$ where $\theta \sim \hat{\pi}$, are upper bounded and lower bounded away from 0. It is an active area of research in random matrix theory to quantify RIP constants e.g. see (Baraniuk et al., 2008).

RIP generalizes to restricted strong convexity and smoothness (Chen & Sanghavi, 2010); thus our results could potentially be extended to general convex $f(\cdot)$ functions. We present our main result next, and defer the details of the theory to section B in the appendix.

Theorem 1. *In the worst case scenario, with Assumption 1, the solutions path found by Automated Accelerated IHT satisfies the following iterative invariant.*

$$\|w_{t+1} - w^*\|_2 \leq \rho|1+\tau_t| \cdot \|w_t - w^*\|_2 + \rho|\tau_t| \cdot \|w_{t-1} - w^*\|_2 + 2\beta_{3k}\sqrt{\beta_{2k}}\|\epsilon\|_2,$$

where $\rho = \left(2\max\{\frac{\beta_{2k}}{\alpha_{3k}} - 1, 1 - \frac{\alpha_{2k}}{\beta_{3k}}\} + \frac{\beta_{4k}-\alpha_{4k}}{\alpha_{3k}}\right)$, *and* $\|\epsilon\|_2 = \|y - \Phi w^*\|_2$ *is the optimal error.*

The theorem provides an upper bound invariant among consecutive iterates of the algorithm. To have a better sense of convergence rate, we can derive linear convergence from our iterative invariant, as shown in Corollary 1.

Corollary 1. *Given the iterative invariant as stated in Theorem 1, and assuming the optimal solution achieves* $\|\epsilon\|_2 = 0$, *the solution found by Algorithm 2 satisfies:*

$$f(w_{t+1}) - f(w^\star) \leq \phi^t \left(\frac{\beta_{2k}}{\alpha_{2k}} f(w_1) + \frac{\rho\tau\beta_{2k}}{\phi\alpha_k} f(w_0)\right),$$

where $\phi = (\rho(1+\tau) + \sqrt{\rho^2(1+\tau)^2 + 4\rho\tau})/2$ *and* $\tau = \max_{i\in[t]} |\tau_i|$. *It is sufficient to show linear convergence to the global optimum, when* $\phi < 1$, *or equivalently* $\rho < 1/(1+2\tau)$.

We note that Theorem 1 holds more generally, and we chose the simplifying condition of $\|\epsilon\|_2 = 0$ for Corollary 1 to clearly highlight the main result of linear convergence. If $\|\epsilon\|_2 > 0$, the linear convergence (up to an error) can be proved in the same way but with more complicated expressions.

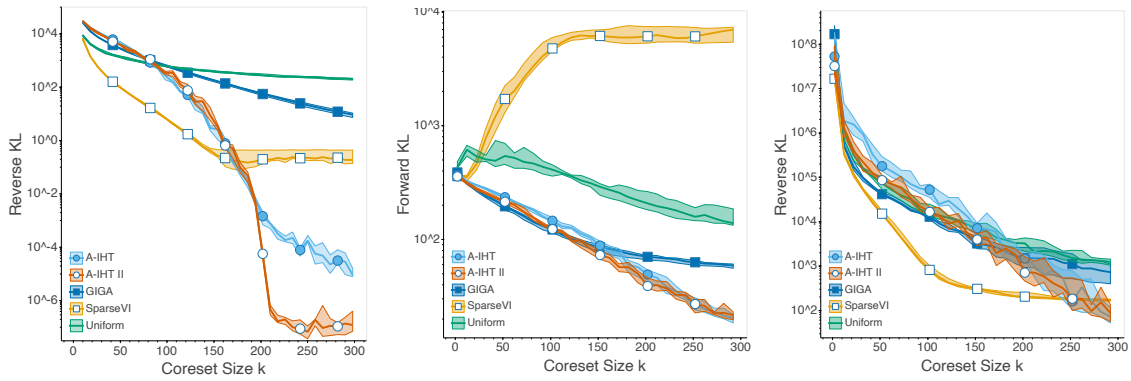
Thus, Algorithm 2 generates a sequence of iterates that decrease the quadratic objective in equation (3) at a geometric rate. The quadratic objective can upper bound the symmetric KL divergence, *i.e.*, the sum of forward KL and reverse KL divergences, between the constructed coreset posterior and the true posterior under certain conditions, as shown in Proposition 2 by Campbell & Beronov (2019), which further justifies our approach of using this objective.

Our theory and algorithm differ from the work by Khanna & Kyrillidis (2018) in several ways. The non-negative constraint is unique to the Bayesian coreset problem, and extending the analysis from the original IHT to our setting is non-trivial (see Section B in appendix). Further, the new analysis we present does not work with the restricted gradient used by Khanna & Kyrillidis (2018), which is why we choose to use the full gradient instead (line 7 in Algorithm 2). We note that the restricted gradient refers to the $\nabla f(z_t)|_{\mathcal{S}}$ in Algorithm 2. We also observe empirically in our experiments that using the full gradient performs better for the coreset problem. The high-level idea is that, during the iterations, it is not guaranteed that $\mathcal{S}$ (line 4 in Algorithm 2) contains the optimal support, while the full gradient is guaranteed to provide information on the optimal support. Further, we also automated the step-size selection, the momentum selection, and the de-bias step selection to minimize the need of tuning. Recall that vanilla IHT (Algorithm 1) is much slower than the greedy approach by Campbell & Broderick (2018), and so the enhancements we propose are crucial to ensure that the overall algorithm is both faster as well as better performing than the state-of-the-art.

4 Related Work

Other scalable approaches for Bayesian inference include subsampling and streaming methods for variational Bayes (Hoffman et al., 2013; Broderick et al., 2013), subsampling methods for MCMC (Welling & Teh, 2011; Ahn et al., 2012; Korattikara et al., 2014; Maclaurin & Adams, 2015), and consensus methods for MCMC (Srivastava et al., 2015; Rabinovich et al., 2015; Scott et al., 2016). These algorithms are motivated by empirical performance and come with few or no theoretical optimization-based guarantees on the inference quality, and often do not scale to larger datasets. Bayesian coresets could be used as part of these approaches, thus resulting into a universal tool for approximate MCMC and variational inference. Recently, Bayesian coresets have been applied to complex models and data. For example, Pinsler et al. (2019) apply Bayesian coresets to batch active learning on Bayesian neural networks with real-world image datasets.

There have been few studies that study convergence properties of approximate inference algorithms. Campbell & Beronov (2019) presented a linear convergence rate, but the assumptions they make are non-standard as the rate of convergence depends on the how well individual samples correlate with the overall loss. Approximation guarantees in terms of KL-divergence are provided (Koyejo et al., 2014; Khanna et al., 2017) for structured sparse posterior inference using the greedy forward selection procedure. Locatello et al. (2017, 2018) study convergence rates for a boosting based algorithm for iteratively refined variational inference.



(a) (b)

Figure 1: (a): Bayesian coresets for synthetic Gaussian posterior inference. (b): Experiments on Bayesian radial basis function regression, with the difference between true posterior and coreset posterior measured in both forward KL and reverse KL. For both (a) and (b), k is the sparsity setting, and the solid lines are the median KL divergence between the constructed coreset posterior and true posterior over 10 trials. The shaded area is the KL divergence between 25^{th} and 75^{th} percentiles.

Thresholding based optimization algorithms have been attractive alternatives to relaxing the constraint to a convex one or to greedy selection. Bahmani et al. (2013) provide a gradient thresholding algorithm that generalizes pursuit approaches for compressed sensing to more general losses. Yuan et al. (2018) study convergence of gradient thresholding algorithms for general losses. Jain et al. (2014) consider several variants of thresholding based algorithms for high dimensional sparse estimation. Additional related works are discussed in Section D in the appendix.

5 Experiments

We empirically examine the performance of our algorithms to construct coresets for Bayesian posterior approximation. Three sets of experiments are presented: Gaussian posterior inference, Bayesian radial basis function regression, and Bayesian logistic and Poisson regression using real-world datasets.

Besides the Automated Accelerated IHT (Algorithm 2), we propose Automated Accelerated IHT - II (Algorithm 3 in section A of appendix), that adds a de-bias step that further improves Algorithm 2 in practice. We refer to the appendix for detailed explanation and discussion of Algorithm 3 due to space limitation.

The proposed algorithms, Automated Accelerated IHT (A-IHT) and Automated Accelerated IHT II (A-IHT II), are compared with three baseline algorithms, *i.e.*, Random (Uniform), Greedy Iterative Geodesic Ascent (GIGA) (Campbell & Broderick, 2018) and Sparse Variational Inference (SparseVI) (Campbell & Beronov, 2019). We use the public Github resources of GIGA and SparseVI for their implementation, where details are provided in our Github repository (link on page 2). We note that the Frank-Wolfe (FW) method proposed in (Campbell & Broderick, 2019) has been shown to be inferior to GIGA and SparseVI in the two corresponding articles, and thus we believe that comparing with GIGA and SparseVI is sufficient.

We calculate the Kullback–Leibler (KL) divergence between the constructed coresets posterior π_w and the true posterior π. We measure both the forward KL divergence $D_{\mathrm{KL}}(\pi\|\pi_w)$ and reverse KL divergence $D_{\mathrm{KL}}(\pi_w\|\pi)$. Both A-IHT and A-IHT II require minimal tuning, *i.e.*, only the stoping criterion is required: $\|w_t - w_{t-1}\| \leq 10^{-5}\|w_t\|$, or number of iterations > 300 for both A-IHT and A-IHT II .

5.1 Synthetic Gaussian posterior inference

We examine the algorithms in this experiment where we have closed-form exact expressions. Specifically, we compare each of these algorithms in terms of optimization accuracy without errors from sampling.

For the D-dimensional Gaussian distribution, we set the parameter $\theta \sim \mathcal{N}(\mu_0, \Sigma_0)$ and draw N i.i.d. samples $x_n \sim \mathcal{N}(\theta, \Sigma)$, which results in a Gaussian posterior distribution with closed-form parameters, as shown in (Campbell & Beronov, 2019). We set the dimension $D = 200$, number of samples $N = 600$, and maximal sparsity k is set to be $1, \ldots, 300$. The initial mean $\mu_0 = 0$, and the initial covariance matrix is set to be $\Sigma_0 = \Sigma = I$. The learning rate for SparseVI is $\gamma_t = 1/t$, and the number of weight update iterations for Sparse VI is 100, as suggested by their paper.

Comparison among all the 5 algorithms measuring the reverse KL divergence between the true posterior

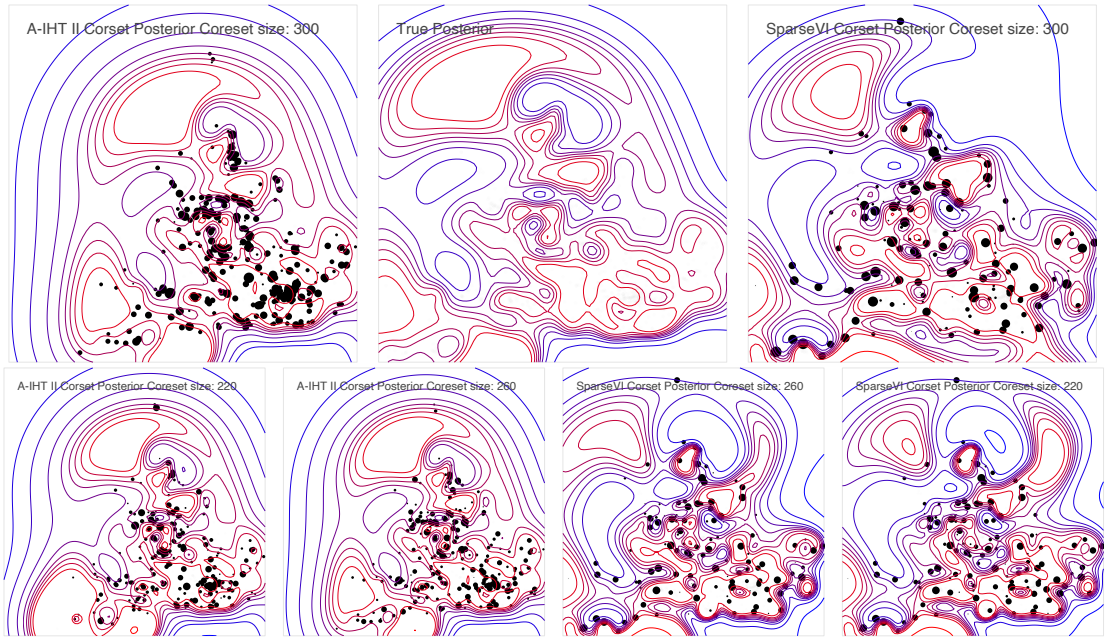


Figure 2: Experiments on Bayesian radial basis function regression, where coreset sparsity setting $k = 220, 260, 300$. Coreset points are presented as black dots, with their radius indicating assigned weights. When $k = 300$, posterior constructed by Accelerated IHT II (top left) shows almost exact contours as the true posterior (top middle), while posterior constructed by SparseVI (top right) shows deviated contours from the true posterior distribution.

and the coreset posterior is presented in Figure 1 (a), which shows that IHT outperforms SparseVI and GIGA, achieving nearly optimal results. We observe that SparseVI stops improving once it hits certain sparsity level, which we suspect is due to the limitations of its greedy nature. It can also be observed that A-IHT II converges faster than A-IHT. Additional results are put in the section E in appendix.

5.2 Bayesian Radial Basis Function Regression

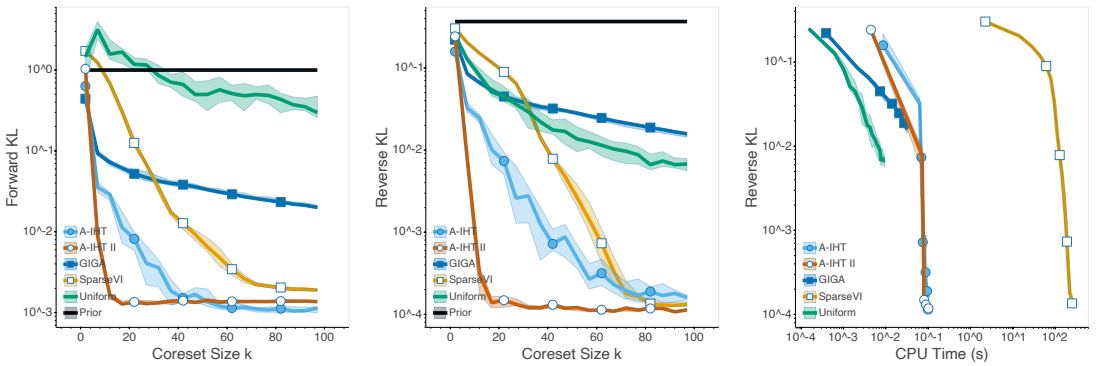
In this subsection, we explore the performance of proposed methods versus the baselines in terms of the both forward KL and reverse KL divergence. The SparseVI algorithm optimizes reverse KL; we show this does not always imply reduction in the forward KL. Indeed selecting more points to greedily optimizing the reverse KL can cause an increase in the forward KL.

We aim to infer the posterior for Bayesian radial basis function regression. Given the dataset[2] $\{(x_n, y_n) \in \mathbb{R}^2 \times \mathbb{R}\}_{n=1}^N$, where x_n is the latitude/longitude coordinates and y_n is house-sale log-price in the United Kingdom, the goal is to infer coefficients $\alpha \in \mathbb{R}^D$ for D radial basis functions $b_d(x) = \exp(-\frac{1}{2\sigma_d^2}(x - \mu_d)^2)$ for $d \in [D]$. The model is $y_n = b_n^\top \alpha + \epsilon_n$, where $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$ with σ^2 be the variance of $\{y_n\}$, and $b_n = [b_1(x_n), \ldots, b_D(x_n)]^\top$. We set prior $\alpha \sim \mathcal{N}(\mu_0, \sigma_0^2 I)$, where μ_0, σ_0^2 are empirical mean and second moment of the data. We subsampled the dataset uniformly at random to $N = 1000$ records for the experiments, and generated 50 basis functions for each of the 6 scales $\sigma_d \in \{0.2, 0.4, 0.8, 1.2, 1.6, 2.0\}$ by generating means μ_d for each basis uniformly from data. Except for the 300 basis functions, an additional near-constant basis of scale 100, with mean corresponding to the mean latitude and longitude of the data, is added. Therefore, $D = 301$ basis functions are considered. Each of the algorithms has access to the closed-form of posterior distribution and covariance (see (Campbell & Beronov, 2019) for detailed derivation).

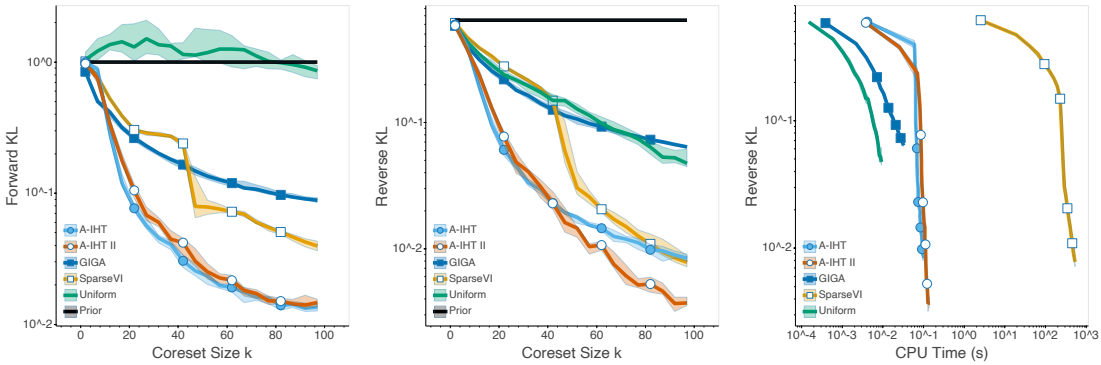
Specific settings for the algorithms are as follows. For SparseVI, the exact covariance can be obtained, and the weight update step can be done without Monte Carlo estimation. For IHT and GIGA, we use true posterior for constructing the ℓ_2 loss function. The learning rate for SparseVI is set to be $\gamma_t = 1/t$, and iteration number $T = 100$, which is the setting SparseVI uses for the experiment (Campbell & Beronov, 2019).

IHT's objective indicates both bounded forward KL and reverse KL. However, SparseVI, which optimizes

[2]The task is to predict housing prices from the UK land registry data (`https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads`) using latitude/longitude coordinates from the Geonames postal code data (`http://download.geonames.org/export/zip/`) as features.



Top Row: `synthetic dataset`



Bottom Row: `phishing dataset`

Figure 3: Bayesian coreset construction for logistic regression (LR) using the `synthetic dataset` (top row) and the `phishing dataset` (bottom row). All the algorithms are run 20 times, and the median as well as the interval of 35^{th} and 65^{th} percentile, indicated as the shaded area, are reported. Different maximal coreset size k is tested from 1 to 100. Forward KL (left column) and reverse KL (middle column) divergence between estimated true posterior and coreset posterior indicate the quality of the constructed coreset. The smaller the KL divergence, the better the coreset is. The running time for each algorithms is also recorded (right column).

the reverse KL, offers no guarantee for the forward KL. As shown in Figure 1 (b), SparseVI increasingly deviates from the true distribution in forward KL as the coreset growths. However, IHT methods offers consistently better coresets in both the metrics.

The reverse KL divergence alone is not enough to indicate good approximation, as shown in Figure 2. We plot the posterior contours for both the true posterior and coreset posterior at a random trial when sparsity level $k = 220, 260, 300$. The coreset posterior constructed by our Algorithm 3 recovers the true posterior almost exactly at $k = 300$, unlike SparseVI. The results for other trials are provided in section F in the appendix.

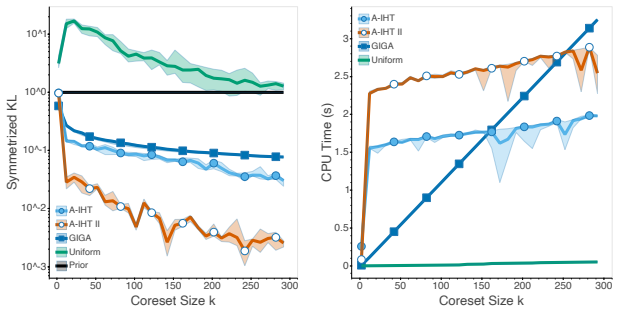
5.3 Bayesian logistic and Poisson regression

We consider how IHT performs when used in real applications where the closed-form expressions are unattainable. Moreover, large-scale datasets are considered to test running time of each algorithm. As the true posterior is unknown, a Laplace approximation is used for GIGA and IHT to derive the finite projection of the distribution, *i.e.*, $\hat{g}_i$. Further, Monte Carlo sampling is used to derive gradients of D_{KL} for SparseVI. We compare different algorithms estimating the posterior distribution for logistic regression and Poisson regression. The reverse KL and forward KL between the coreset posterior and true posterior are estimated using another Laplace approximation. The mode of the Laplace approximation is derived by maximizing the corresponding posterior density. The experiment was proposed by Campbell & Broderick (2019), and is used in (Campbell & Broderick, 2018) and (Campbell & Beronov, 2019). Due to space limitations, we refer to section G in the appendix for details of the experimental setup, and extensive additional results.

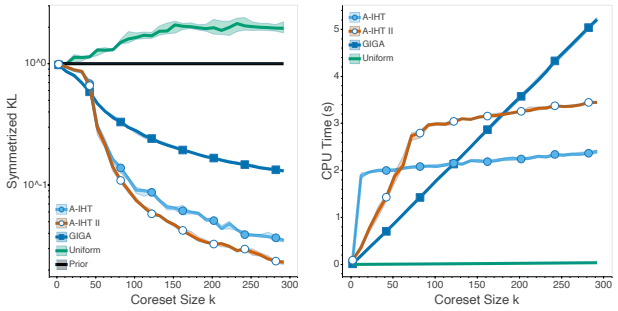
For logistic regression, given a dataset $\{(x_n, y_n) \in \mathbb{R}^D \times \{1, -1\} \mid n \in [N]\}$, we aim to infer $\theta \in \mathbb{R}^{D+1}$ based on the model:

$$y_n \mid x_n, \theta \sim \mathrm{Bern}\left(\frac{1}{1 + e^{-z_n^\top \theta}}\right),$$

where $z_n = [x_n^\top, 1]^\top$. We set $N = 500$ by uniformly



Top Row: `large synthetic dataset`



Bottom Row: `original phishing dataset`

Figure 4: Bayesian coreset construction for logistic regression (LR) using the `large synthetic dataset` (top row) and the `original phishing dataset` (bottom row). All the algorithms are run 10 times, and the median as well as the interval of 35^{th} and 65^{th} percentile, indicated as the shaded area, are reported. Different maximal coreset size k is tested. Symmetrized KL divergence between estimated true posterior and coreset posterior indicate the quality of the constructed coreset (left column). The running time for each algorithms is also recorded (right column).

sub-sampling from datasets due to the high computation cost of SparseVI. Three datasets are used for logistic regression. Two of them are: the `synthetic dataset` consists of x_n sampled i.i.d. from normal distribution $\mathcal{N}(0, I)$, and label y_n sampled from Bernoulli distribution conditioned on x_n and $\theta = [3, 3, 0]^\top$. The `phishing dataset`[3] is preprocessed (Campbell & Beronov, 2019) via PCA to dimension of $D = 10$ to mitigate high computation by SparseVI.

We present two sets of experiments, *i.e.*, logistic regression using the `synthetic dataset` and the `phishing dataset`, in Figure 3. One other set of experiments on logistic regression, and three sets of experiments on Poisson regression are deferred to section G in appendix.

It is observed that A-IHT and A-IHT II achieve state-of-the-art performance. The IHT algorithms often obtain coresets with smaller KL between the coreset posterior and true posterior than GIGA and SparseVI, with computing time comparable to GIGA and significantly less than SparseVI. We conjecture that GIGA and SparseVI perform worse than our methods due to their greedy nature: they can be "short-sighted" and do not rectify past decisions. The experiments indicate that IHT outperforms the previous methods, improving the trade-off between accuracy and performance.

Large-scale Datasets. Two large datasets are considered: *i*) the `large synthetic dataset` for logistic regression is generated following the same procedure as before, but with dataset size $N = 9000$; *ii*) the `original phishing` dataset has size $N = 11055$ and dimension $D = 68$. The maximal iteration number of the two IHT algorithms is 500. Symmetrized KL, *i.e.*, the sum of forward and reverse KL, is reported.

Results are shown in Figure 4. We have to omit SparseVI due to its prohibitively high cost (*e.g.*, as shown in Figure 3, SparseVI needs $\times 10^4$ more time than IHT and GIGA). As our complexity analysis of the algorithms in subsection 3.2, the running time of GIGA grows linearly with respect to the coreset size k, while that is almost free for IHT. GIGA begins to cost more time than IHT at $k \approx 200$, *i.e.*, about only 2% of the dataset.

Additional evaluation. For large-scale datasets, it is often necessary to "batch" the algorithms. We test the performance of IHT using a stochastic gradient estimator. The gradient estimator is calculated with random batches in each iteration, where we use a batch size of 20% of the full dataset size. Results on six datasets are defer to section G in appendix.

Moreover, as an alternative evaluation of the quality of constructed coresets, we test the ℓ_2-distance between the maximum-a-posteriori (MAP) estimation of the full-dataset posterior and coreset posterior. Results on six datasets are deferred to section G in appendix.

6 Conclusion

In this paper, we consider the Bayesian coreset construction problem from a sparse optimization perspective, through which we propose a new algorithm that incorporates the paradigms of sparse as well as accelerated optimization. We provide theoretical analysis for our method, showing linear convergence under standard assumptions. Finally, numerical results demonstrate the improvement in both accuracy and efficiency when compared to the state of the art methods. Our viewpoint of using sparse optimization for Bayesian coresets can potentially help to consider more complex structured sparsity, which is left as future work.

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

Acknowledgements

AK acknowledges funding by the NSF (CCF-1907936, CNS-2003137). AK thanks TOOL's Danny Carey for his percussion performance in "Pneuma". We would like to thank the reviewers for their valuable and constructive comments. Their feedback enables us to further improve the paper.

References

Sungjin Ahn, Anoop Korattikara, and Max Welling. Bayesian posterior sampling via stochastic gradient fisher scoring. In *29th International Conference on Machine Learning, ICML 2012*, pp. 1591–1598, 2012.

Sohail Bahmani, Bhiksha Raj, and Petros T. Boufounos. Greedy sparsity-constrained optimization. *J. Mach. Learn. Res.*, 14(1):807–841, March 2013. ISSN 1532-4435.

Luca Baldassarre, Nirav Bhan, Volkan Cevher, Anastasios Kyrillidis, and Siddhartha Satpathi. Group-sparse model selection: Hardness and relaxations. *IEEE Transactions on Information Theory*, 62(11): 6508–6534, 2016.

Richard Baraniuk, Mark A. Davenport, Ronald A. DeVore, and Michael B. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28:253–263, 2008.

Richard G Baraniuk, Volkan Cevher, Marco F Duarte, and Chinmay Hegde. Model-based compressive sensing. *IEEE Transactions on information theory*, 56 (4):1982–2001, 2010.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2:183–202, 2009.

T. Blumensath and M. Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

Thomas Blumensath. Accelerated iterative hard thresholding. *Signal Process.*, 92(3):752–756, March 2012. ISSN 0165-1684. doi: 10.1016/j.sigpro.2011.09.017. URL `https://doi.org/10.1016/j.sigpro.2011.09.017`.

Thomas Blumensath and Mike E Davies. Normalized iterative hard thresholding: Guaranteed stability and performance. *IEEE Journal of selected topics in signal processing*, 4(2):298–309, 2010.

Tamara Broderick, Nicholas Boyd, Andre Wibisono, Ashia C Wilson, and Michael I Jordan. Streaming variational bayes. In *Advances in Neural Information Processing Systems*, pp. 1727–1735, 2013.

Trevor Campbell and Boyan Beronov. Sparse variational inference: Bayesian coresets from scratch. In *Advances in Neural Information Processing Systems*, pp. 11457–11468, 2019.

Trevor Campbell and Tamara Broderick. Bayesian coreset construction via greedy iterative geodesic ascent. In *International Conference on Machine Learning*, pp. 697–705, 2018.

Trevor Campbell and Tamara Broderick. Automated scalable bayesian inference via hilbert coresets. *The Journal of Machine Learning Research*, 20(1):551–588, 2019.

Emmanuel J Candes. The restricted isometry property and its implications for compressed sensing. *Comptes rendus mathematique*, 346(9-10):589–592, 2008.

Yuxin Chen and Sujay Sanghavi. A general framework for high-dimensional estimation in the presence of incoherence. In *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1570–1576. IEEE, 2010.

Wei Dai and Olgica Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE transactions on Information Theory*, 55(5):2230–2249, 2009.

David L Donoho et al. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

Simon Foucart. Hard thresholding pursuit: an algorithm for compressive sensing. *SIAM Journal on Numerical Analysis*, 49(6):2543–2563, 2011.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

E. Ghadimi, H. R. Feyzmahdavian, and M. Johansson. Global convergence of the heavy-ball method for convex optimization. In *2015 European Control Conference (ECC)*, pp. 310–315, July 2015.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1): 1303–1347, 2013.

Jonathan Huggins, Trevor Campbell, and Tamara Broderick. Coresets for scalable bayesian logistic regression. In *Advances in Neural Information Processing Systems*, pp. 4080–4088, 2016.

Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th international conference on machine learning*, pp. 427–435, 2013.

Prateek Jain, Ambuj Tewari, and Purushottam Kar. On iterative hard thresholding methods for high-dimensional m-estimation. In *Proceedings of the 27th*

International Conference on Neural Information Processing Systems - Volume 1, NIPS'14, pp. 685–693, Cambridge, MA, USA, 2014. MIT Press.

Prateek Jain, Nikhil Rao, and Inderjit S Dhillon. Structured sparse regression via greedy hard thresholding. In *Advances in Neural Information Processing Systems*, pp. 1516–1524, 2016.

Rajiv Khanna and Anastasios Kyrillidis. Iht dies hard: Provable accelerated iterative hard thresholding. In *International Conference on Artificial Intelligence and Statistics*, pp. 188–198, 2018.

Rajiv Khanna, Joydeep Ghosh, Rusell Poldrack, and Oluwasanmi Koyejo. Information projection and approximate inference for structured sparse variables. In *Artificial Intelligence and Statistics*, pp. 1358–1366, 2017.

Anoop Korattikara, Yutian Chen, and Max Welling. Austerity in mcmc land: Cutting the metropolis-hastings budget. In *International Conference on Machine Learning*, pp. 181–189, 2014.

Oluwasanmi Koyejo, Rajiv Khanna, Joydeep Ghosh, and Russell A. Poldrack. On prior distributions and approximate inference for structured variables. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pp. 676–684, Cambridge, MA, USA, 2014. MIT Press.

Anastasios Kyrillidis and Volkan Cevher. Recipes on hard thresholding methods. In *2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 353–356. IEEE, 2011.

Anastasios Kyrillidis and Volkan Cevher. Matrix recipes for hard thresholding methods. *Journal of Mathematical Imaging and Vision*, 2(48):235–265, 2014.

Anastasios Kyrillidis, Luca Baldassarre, Marwa El Halabi, Quoc Tran-Dinh, and Volkan Cevher. Structured sparsity: Discrete and convex approaches. In *Compressed Sensing and its Applications*, pp. 341–387. Springer, 2015.

Xingguo Li, Tuo Zhao, Raman Arora, Han Liu, and Jarvis Haupt. Stochastic variance reduced optimization for nonconvex sparse learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 917–925, New York, New York, USA, 20–22 Jun 2016. PMLR.

Francesco Locatello, Rajiv Khanna, Joydeep Ghosh, and Gunnar Rätsch. Boosting variational inference: an optimization perspective. In *AISTATS*, 2017.

Francesco Locatello, Gideon Dresdner, Rajiv Khanna, Isabel Valera, and Gunnar Rätsch. Boosting black box variational inference. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 3405–3415, Red Hook, NY, USA, 2018. Curran Associates Inc.

Yi-An Ma, Yuansi Chen, Chi Jin, Nicolas Flammarion, and Michael I. Jordan. Sampling can be faster than optimization. *Proceedings of the National Academy of Sciences*, 116(42):20881–20885, 2019. ISSN 0027-8424. doi: 10.1073/pnas.1820003116.

Dougal Maclaurin and Ryan Prescott Adams. Firefly monte carlo: Exact mcmc with subsets of data. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

Deanna Needell and Joel A Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and computational harmonic analysis*, 26(3):301–321, 2009.

Yurii E Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. In *Dokl. akad. nauk Sssr*, volume 269, pp. 543–547, 1983.

Nam Nguyen, Deanna Needell, and Tina Woolf. Linear convergence of stochastic iterative greedy algorithms with sparse constraints. *IEEE Transactions on Information Theory*, 63:6869–6895, 2014.

Robert Pinsler, Jonathan Gordon, Eric Nalisnick, and José Miguel Hernández-Lobato. Bayesian batch active learning as sparse subset approximation. *Advances in Neural Information Processing Systems*, 32: 6359–6370, 2019.

Maxim Rabinovich, Elaine Angelino, and Michael I Jordan. Variational consensus monte carlo. In *Advances in Neural Information Processing Systems*, pp. 1207–1215, 2015.

Steven L Scott, Alexander W Blocker, Fernando V Bonassi, Hugh A Chipman, Edward I George, and Robert E McCulloch. Bayes and big data: The consensus monte carlo algorithm. *International Journal of Management Science and Engineering Management*, 11(2):78–88, 2016.

Shai Shalev-Shwartz, Nathan Srebro, and Tong Zhang. Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM J. on Optimization*, 20(6):2807–2832, August 2010. ISSN 1052-6234. doi: 10.1137/090759574.

Sanvesh Srivastava, Volkan Cevher, Quoc Dinh, and David Dunson. Wasp: Scalable bayes via barycenters of subset posteriors. In *Artificial Intelligence and Statistics*, pp. 912–920, 2015.

Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Xiao-Tong Yuan, Ping Li, and Tong Zhang. Gradient hard thresholding pursuit. *Journal of Machine Learning Research*, 18(166):1–43, 2018.

Jacky Y Zhang, Rajiv Khanna, Anastasios Kyrillidis, and Oluwasanmi O Koyejo. Learning sparse distributions using iterative hard thresholding. In *Advances in Neural Information Processing Systems 32*, pp. 6757–6766. Curran Associates, Inc., 2019.