Learning Sparse Distributions using Iterative Hard Thresholding

Jacky Y. Zhang
Department of Computer Science
University of Illinois at Urbana-Champaign
yiboz@illinois.edu

Rajiv Khanna
Department of Statistics
University of California at Berkeley
rajivak@berkeley.edu

Anastasios Kyrillidis
Department of Computer Science
Rice University
rajivak@berkeley.edu

Oluwasanmi Koyejo
Department of Computer Science
University of Illinois at Urbana-Champaign
sanmi@illinois.edu

Abstract

Iterative hard thresholding (IHT) is a projected gradient descent algorithm, known to achieve state of the art performance for a wide range of structured estimation problems, such as sparse inference. In this work, we consider IHT as a solution to the problem of learning sparse discrete distributions. We study the hardness of using IHT on the space of measures. As a practical alternative, we propose a greedy approximate projection which simultaneously captures appropriate notions of sparsity in distributions, while satisfying the simplex constraint, and investigate the convergence behavior of the resulting procedure in various settings. Our results show, both in theory and practice, that IHT can achieve state of the art results for learning sparse distributions.

1 Introduction

Probabilistic models provide a flexible approach for capturing uncertainty in real world processes, with a variety of applications which include latent variable models and density estimation, among others. Like other machine learning tools, probabilistic models can be enhanced by encouraging parsimony, as this captures useful inductive biases. In practice, this often improves the interpretability and generalization performance of the resulting models, and is particularly useful in applied settings with limited samples compared to the model degrees of freedom. One of the most effective parsimonious assumptions is sparsity. As such, learning sparse distributions is a problem of broad interest in machine learning, with many applications [1–7].

The majority of approaches for sparse probabilistic modeling have focused on the construction of appropriate priors based on inputs from domain experts. The technical challenges there involve the challenges of prior design and inference [3, 1, 8], including methods that are additionally designed to exploit special structures [5, 4, 7] . More recently, there has been an interest in studying these algorithmic approaches from an optimization perspective [9–11], with the goal of a deeper understanding and, in some cases, even suggesting improvements over previous methods [12, 13].In this work, we consider an optimization-based approach to learning sparse discrete distributions. Despite wide applicability, when compared to classical constrained optimization, there are limited studies that focus on the understanding, both in theory and in practice, of optimization methods over the space of probability densities, under sparsity constraints.

Our present work proposes and investigates the use of Iterative Hard Thresholding (IHT [14–18]) for the problem of sparse probabilistic estimation. IHT is an iterative algorithm that is well-studied

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

in the classical optimization literature. Further, there are known worst-case convergence guarantees and empirical studies [19, 20] that vouch for its performance. Our goal in this work is to investigate the convergence properties of IHT, when applied to probabilistic densities, and to evaluate its efficacy for learning sparse distributions.

However, transferring this algorithm from vector and matrix spaces to the space of measures is not straightforward. While several of the technical pieces –such as the existence of a variational derivative and normed structure– fall into place, the algorithm is an iterative one, that involves solving a projection subproblem in each iteration. We show that this subproblem is computationally hard in general, but provide an approximate procedure that we analyze under certain assumptions.

Our contributions in this work are algorithmic and theoretical, with proof of concept empirical evaluation. We briefly summarize our contributions below.

- We propose the use of classical IHT for learning sparse distributions, and show that the space of measures meets the structural requirements for IHT.
- We study in depth the hardness of the projection subproblem, showing that it is NP-hard, and no polynomial-time algorithm exists that can solve it with guarantees.
- Since the projection problem is solved in every iteration, we propose a simple greedy algorithm and provide sufficient theoretical conditions, under which the algorithm provably approximates the otherwise hard projection problem.
- We draw on techniques from classical optimization to provide convergence rates for the overall IHT algorithm: i.e., we study after how many iterations will the algorithm guarantee to be within some small ϵ of the true optimum.

In addition to our conceptual and theoretical results, we present empirical studies that support our claims.

2 Problem statement

Preliminaries. We use bold characters to denote vectors. Given a vector $\boldsymbol{v}$, we use v_i to represent its i-th entry. We use calligraphic upper case letters to denote sets; *e.g.*, $\mathcal{S}$. With a slight abuse of notation, we will use lower case letters to denote probability distributions *e.g.*, p, q, as well as functions *e.g.*, f. The distinction from scalars will be apparent from the context; we usually append functions with parentheses to distinguish from scalars. We use upper case letters to denote functionals *i.e.*, functions that take as an input other functions *e.g.*, $F[p(\cdot)]$. We use $[n]$ to denote the set $\{1, 2, ...n\}$. Given a set of indices $\mathcal{S} \subset [n]$, we denote the cardinality of $\mathcal{S}$ as $|\mathcal{S}|$. Given a vector $\boldsymbol{x}$, we denote its support set *i.e.*, the set of non-zero entries, as supp$(\boldsymbol{x})$. We use $\mathbb{P}\{e\}$ to denote the probability of event e.

Let $\mathcal{P}$ denote the set of discrete n-dimensional probability densities on an n-dimensional domain $\mathcal{X}$:

$$\mathcal{P} = \left\{ p(\cdot) \, : \mathcal{X} \to \mathbb{R}_+ \mid \sum_{\boldsymbol{x} \in \mathcal{X}} p(\boldsymbol{x}) = 1 \right\}.$$

Let $\mathcal{S} \subset [n]$ denote a support set where $|\mathcal{S}| = k < n$. Let $\mathcal{X}_\mathcal{S} \subset \mathcal{X}$ denote the set of variables with support $\mathcal{S}$, *i.e.*,

$$\mathcal{X}_\mathcal{S} = \big\{ \boldsymbol{x} \in \mathcal{X} \mid \text{supp}(\boldsymbol{x}) \subseteq \mathcal{S} \big\}.$$

The set of domain restricted densities, denoted by $\mathcal{P}_\mathcal{S}$, is the set of probability density functions supported on $\mathcal{X}_\mathcal{S}$; *i.e.*,

$$\mathcal{P}_\mathcal{S} = \{ q(\cdot) \in \mathcal{P} \mid \forall \boldsymbol{x} \notin \mathcal{X}_\mathcal{S}, \, q(\boldsymbol{x}) = 0 \}.$$

Inversely, we denote the support of a domain restricted density $q(\cdot) \in \mathcal{P}_\mathcal{S}$ as supp$(q) = \mathcal{S}$. Next, we define the notion of sparse distributions.

Definition 1 (Distribution Sparsity [5]). *Let $\mathcal{D}_k = \cup_{|\mathcal{S}| \leq k} \mathcal{P}_\mathcal{S} \subseteq \mathcal{P}$* i.e.*, the union of all possible k-sparse support domain restricted densities. We say that $p(\cdot)$ is k-sparse if $p(\cdot) \in \mathcal{D}_k$.*

Note that while each component $\mathcal{P}_\mathcal{S}$ is a convex set, the union $\mathcal{D}_k$ is not. To see this, consider the convex combination of two k-sparse distributions p_1 and p_2 with disjoint supports $\mathcal{S}_1$ and $\mathcal{S}_2$

respectively. In general, the convex combination $\alpha p_1(\cdot) + (1-\alpha)p_2(\cdot);\ 0 < \alpha < 1$, has larger support; i.e., $|\mathcal{S}_1 \cup \mathcal{S}_2| > k$. As an aside, we note that unlike the vector case, its is straightforward to construct multiple definitions of distribution sparsity. For instance, another reasonable definition is via the set $\mathcal{D}'_k = \{p(\cdot) \in \mathcal{P} \mid p(\boldsymbol{x}) = 0 \text{ for all } \|\boldsymbol{x}\|_0 > k\}$; i.e., distributions that assign zero probability mass to non-k-sparse vectors. Interestingly, $\mathcal{D}_k \subset \mathcal{D}'_k \subset \mathcal{P}$ in general, as any of the distributions in $\mathcal{D}_k$ must has a support with size less than k, which is not necessary for distributions in $\mathcal{D}'_k$. Motivated by prior work [5], we use Definition 1 in this work.

Vector sparsity. While the proposed framework is developed for a specialized notion of sparsity i.e. along the dimensions of a multivariate discrete distribution, it is also applicable to alternative notions of distribution sparsity. One common setting is sparsity of the distribution itself $p(\cdot)$ when represented as a vector e.g. sparsifying the number of valid states of a univariate distribution such as a histogram. We outline how our framework can be applied to this setting in the Appendix A.

Problem setting. In this work, we focus on studying sparsity for the case of discrete densities. In particular, $\mathcal{X} \subset \mathbb{Z}^n$; i.e., $\boldsymbol{x}$ is an integer such that:

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{Z}^n \mid \forall i \in [n], 0 \leq x_i \leq m-1\},$$

where m is an integer. Therefore, $\boldsymbol{x}$ has m^n valid positions. In other words, if we denote X as a random variable from that distribution, then $X \in \mathcal{X}$ has m^n possible values, and $\mathbb{P}\{X = \boldsymbol{x}\} = p(\boldsymbol{x})$.

Given a cost functional over distributions $F[\cdot] : \mathcal{P} \to \mathbb{R}$, we are interested in the following optimization criterion:

$$\min_q \quad F[q] \quad \text{subject to} \quad q \in \mathcal{D}_k, \tag{1}$$

where $\mathcal{D}_k = \cup_{\mathcal{S}:|\mathcal{S}|\leq k}\mathcal{P}_\mathcal{S} \subseteq \mathcal{P}$ is the k-sparsity constraint, as in Definition 1. In words, we are interested in finding a *distribution*, denoted as $q(\cdot)$, that "lives" in the k-sparse set of distributions, and minimizes the cost functional $F[\cdot]$. This is similar to classical *sparse optimization* problems in literature [21–24], but there are fundamental difficulties, both in theory and in practice, that require a different approach than standard iterative hard thresholding algorithms [14–18].

We assume that the objective $F[\cdot]$ is a *convex* functional over distributions.

Definition 2 (Convexity of $F[\cdot]$). *The functional $F[\cdot] : \mathcal{P} \to \mathbb{R}$ is convex if:*

$$F[\theta q(\cdot) + (1-\theta)p(\cdot)] \leq \theta F[q(\cdot)] + (1-\theta)F[p(\cdot)],$$

for all $q(\cdot),\ p(\cdot) \in \mathcal{P}$ and $\theta \in [0,1]$.

Observe that, while $F[\cdot]$ is a convex functional, and $\mathcal{P}$ and $\mathcal{P}_\mathcal{S}$ are convex sets, $\mathcal{D}_k$ is not a convex set. Hence, the optimization problem (1) is not a convex program.

Following the projected gradient descent approach, we require definitions of the gradient over $F[\cdot]$, as well as definitions of the projection.

Definition 3 (Variational Derivative [25]). *The variational derivative of $F[\cdot] : \mathcal{P} \to \mathbb{R}$ is a function, denoted as $\frac{\delta F}{\delta q}(\cdot) : \mathcal{X} \to \mathbb{R}$, and satisfies:*

$$\sum_{\mathcal{X}} \tfrac{\delta F}{\delta q}(\boldsymbol{x})\phi(\boldsymbol{x}) = \tfrac{\partial F[q+\epsilon\phi]}{\partial \epsilon}\Big|_{\epsilon=0}$$

where $\phi : \mathcal{X} \to \mathbb{R}$ is an arbitrary function.

Definition 4 (First-order Convexity). *The functional $F[\cdot] : \mathcal{P} \to \mathbb{R}$ is convex if:*

$$F[q(\cdot)] \geq F[p(\cdot)] + \left\langle \tfrac{\delta F}{\delta p}(\cdot), q(\cdot) - p(\cdot) \right\rangle$$

for all $q(\cdot),\ p(\cdot) \in \mathcal{P}$.

Here, we use the standard inner product for two densities: $\langle q(\cdot), p(\cdot)\rangle = \int_x q(x)p(x)$, or $\langle q(\cdot), p(\cdot)\rangle = \sum_x q(x)p(x)$ in the discrete setting.

3 Algorithms

Recall that our goal is to solve the optimization problem (1). A natural way to solve it in an iterative fashion is using *projected gradient descent*, where the projection step is over the set of sparse distributions $\mathcal{D}_k$. This analogy makes the connection to iterative hard thresholding (IHT) algorithms, where the iterative recursion is:

$$p_{t+1}(\cdot) = \Pi_{\mathcal{D}_k}\left(p_t(\cdot) - \mu \tfrac{\delta F}{\delta p_t}(\cdot)\right),$$

where $p_t(\cdot)$ denotes the current iterate, and $\Pi_{\mathcal{D}_k}(\cdot)$ denotes, in an abstract sense, the projection of the distribution function to the set of sparse distribution functions. The consequent steps are analogous to those of regular IHT: given an initialization point, we iteratively $i)$ compute the gradient, $ii)$ perform the gradient step with step size μ, $iii)$ ensure the computed approximate solution satisfies our constraint in each iteration by projecting to $\mathcal{D}_k$.

Algorithm 1 Distribution IHT

```
1: **Input:** $F[\cdot] : \mathcal{P} \to \mathbb{R}$, $k \in \mathbb{Z}_+$. number of
   iters $T$, $p_0(\cdot) \in \mathcal{D}_k$, $\mu$. **Output:** $p_T \in \mathcal{D}_k$
2: $t \leftarrow 0$
3: **while** $t < T$ **do**
4:     $q_{t+1}(\cdot) = p_t(\cdot) - \mu \frac{\delta F}{\delta p_t}(\cdot)$
5:     $p_{t+1}(\cdot) = \Pi_{\mathcal{D}_k}(q_{t+1})$
6: **end while**
7: **return** $p_T(\cdot)$
```

3.1 Projection onto $\mathcal{D}_k$

Consider the projection step with respect to the ℓ_2-norm i.e.

$$\Pi_{\mathcal{D}_k}(p(\cdot)) := \operatorname*{arg\,min}_{q(\cdot)\in\mathcal{D}_k} \|q(\cdot) - p(\cdot)\|_2^2, \quad (2)$$

where the ℓ_2-norm is defined by the aforementioned inner product $\langle q(\cdot), p(\cdot)\rangle = \sum_{\boldsymbol{x}} q(\boldsymbol{x})p(\boldsymbol{x})$. The set $\mathcal{D}_k = \cup_{|\mathcal{S}|\le k}\mathcal{P}_{\mathcal{S}}$ is a union of $\binom{n}{k} = O(n^k)$ sparse sets $\mathcal{P}_S$ of different supports. Thus, if we denote $\mathtt{T}_{\text{proj}}$ as the time to compute $\Pi_{\mathcal{P}_S}(p(\cdot))$, then we need $O(n^k \cdot \mathtt{T}_{\text{proj}})$ time for $\mathcal{D}_k$ projection using naive enumeration. One may reasonably conjecture the existence of more efficient implementations of the exact projection in (2), e.g., in polynomial time. In the following, we show that this is not the case.

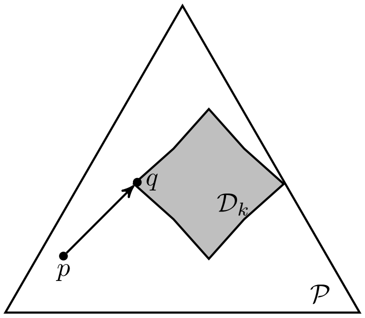


Figure 1: Illustration of projection onto $\mathcal{D}_k$, with $q = \Pi_{\mathcal{D}_k}(p)$.

3.2 On the tractability of sparse distribution ℓ_2-norm projection

The projection (2) is iteratively solved in IHT (step 5 in Algorithm 1). Thus, for the algorithm to be practical, it is important to study the tractability of the projection step. The combinatorial nature of $\mathcal{D}_k$ hints that this might not be the case.

Theorem 1. *The sparse distribution ℓ_2-norm projection problem* (2) *is NP-hard.*

Sketch of proof: We show that the subset selection problem [26] can be reduced to the ℓ_2-norm projection problem. The complete proof is provided in the supplementary material.

As an alternative route, NP-hard problems can be often tackled sufficiently, by using approximate methods. However, the following theorem states that the sparsity constrained optimization problem in (2) is hard even to approximate, in the sense that no deterministic approximation algorithm exists that solves it in polynomial time.

Theorem 2. *There exists no deterministic algorithm that can provide a constant factor approximation for the sparse distribution ℓ_2-norm projection problem in polynomial time. Formally, for given $q : \mathcal{X} \to \mathbb{R}$ with $\mathcal{X} \in \mathbb{R}^n$, let $p^\star(\cdot)$ be the optimal ℓ_2-norm projection onto $\mathcal{D}_k$, and let $\widehat{p}(\cdot)$ be the solution found by any algorithm that operates in $O(poly(n))$ time. Then, we can design problem instances, where the approximation ratio:*

$$\varphi = \frac{\|q(\cdot) - \widehat{p}(\cdot)\|_2^2}{\|q(\cdot) - p^\star(\cdot)\|_2^2} - 1,$$

cannot be bounded.

The proof of the theorem is provided in the supplementary material. Through Theorems 1 and 2, we have shown that the distribution sparse ℓ_2-norm projection problem is hard, and thus the applicability of IHT on the space of densities seems not to be well-established to be practical. This may be surprising, in light of results in a variety of domains where it is known to be effective. For example, in case of vectors, a simple $O(n)$ selection algorithm solves the projection problem *optimally* [27]. Similarly, on the space of matrices for low rank IHT, the projection onto the top-k ranks is optimally solved by an SVD [28].

3.3 A greedy approximation

In contrast to the results of Theorems 1 and 2, we have observed that a simple greedy support selection seems effective in practice. Thus, we simply consider replacing exact projection to $\mathcal{D}_k$ by greedy selection.

Algorithm 2 Greedy Sparse Projection (GSProj)

```
1: Input: $n$-dimensional function $q : \mathcal{X} \to \mathbb{R}$ and sparsity level $k$.
2: Output: A distribution $p(\cdot) \in \mathcal{D}_k$
3: $\mathcal{S} := \emptyset$
4: while $|\mathcal{S}| < k$ do
5:     $j \in \arg\min_{i \in [n] \setminus \mathcal{S}} \left\{ \min_{p \in \mathcal{P}_{\mathcal{S} \cup i}} \|p(\cdot) - q(\cdot)\|_2^2 \right\}$
6:     $\mathcal{S} := \mathcal{S} \cup j$
7: end while
8: return $\arg\min_{p \in \mathcal{P}_{\mathcal{S}}} \|p(\cdot) - q(\cdot)\|_2^2$
```

Consider Algorithm 2 when the input is not necessarily a distribution, *i.e.*, $\sum_{\boldsymbol{x} \in \mathcal{X}} q(\boldsymbol{x}) \neq 1$. The key procedure of the projection is line 5, where the inner $\min(\cdot)$ is the projection of $q(\cdot)$ on a set of domain restricted densities. Let $\widehat{p}(\cdot)$ denote this projection, *i.e.*, $\widehat{p}(\cdot) = \arg\min_{p(\cdot) \in \mathcal{P}_{\mathcal{S}}} \|p(\cdot) - q(\cdot)\|_2^2$. Since, by definition $\widehat{p}(\boldsymbol{x}) = 0$ for any $\boldsymbol{x} \notin \mathcal{X}_{\mathcal{S}}$, we only need to calculate $\widehat{p}(\boldsymbol{x})$ where $\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}}$, and this can be reformulated as:

$$\arg\min_{p(\cdot)} \sum_{\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}}} (p(\boldsymbol{x}) - q(\boldsymbol{x}))^2 \qquad \text{s.t.} \sum_{\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}}} p(\boldsymbol{x}) = 1 \quad \text{and} \quad \forall_{\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}}} p(\boldsymbol{x}) \geq 0,$$

which is essentially ℓ_2-norm projection onto a simplex $\{p(\boldsymbol{x}) \mid \sum_{\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}}} p(\boldsymbol{x}) = 1, \forall_{\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}}} p(\boldsymbol{x}) \geq 0\}$. This ℓ_2-norm projection onto the simplex can be solved efficiently and easily (See [29]).

When $p(\cdot)$ is a distribution, we can analytically compute its projection on any support restricted domain. Given support $\mathcal{S}$, the exact projection of a distribution $p(\cdot)$ onto $\mathcal{P}_{\mathcal{S}}$ is:

$$\arg\min_{q \in \mathcal{P}_{\mathcal{S}}} \|q(\cdot) - p(\cdot)\|_2^2. \tag{3}$$

In our setting, the above problem can be written as

$$\begin{aligned} \arg\min_{q \in \mathcal{P}_{\mathcal{S}}} \|q(\cdot) - p(\cdot)\|_2^2 &= \arg\min_{q \in \mathcal{P}_{\mathcal{S}}} \langle q(\cdot) - p(\cdot), q(\cdot) - p(\cdot) \rangle = \arg\min_{q \in \mathcal{P}_{\mathcal{S}}} \sum_{\boldsymbol{x} \in \mathcal{X}} (q(\boldsymbol{x}) - p(\boldsymbol{x}))^2 \\ &= \arg\min_{q \in \mathcal{P}_{\mathcal{S}}} \sum_{\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}}} (q(\boldsymbol{x}) - p(\boldsymbol{x}))^2 + \sum_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{x} \notin \mathcal{X}_{\mathcal{S}}} p(\boldsymbol{x})^2. \end{aligned}$$

The last equation is due to definition of $\mathcal{P}_{\mathcal{S}}$ and $\mathcal{X}_{\mathcal{S}}$. Since $p(\cdot)$ is constant, we can eliminate the last term. Further, since $q \in \mathcal{P}_{\mathcal{S}}$, we have that $q(\boldsymbol{x}) = 0$ for every $\boldsymbol{x} \notin \mathcal{X}_{\mathcal{S}}$. The resulting problem is:

$$\arg\min_{q \in \mathcal{P}_{\mathcal{S}}} \sum_{\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}}} (q(\boldsymbol{x}) - p(\boldsymbol{x}))^2 \quad \text{s.t.} \quad \sum_{\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}}} q(\boldsymbol{x}) = 1. \tag{4}$$

Denote $\sum_{\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}}} p(\boldsymbol{x}) = C \leq 1$. Applying the Quadratic Mean-Arithmetic Mean inequality to equation (4), we have:

$$\sum_{\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}}} (q(\boldsymbol{x}) - p(\boldsymbol{x}))^2 \geq (1 - C)^2 / |\mathcal{X}_{\mathcal{S}}| \quad \text{s.t.} \quad \sum_{\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}}} q(\boldsymbol{x}) = 1$$

The equality can be achieved when $q(\boldsymbol{x}) - p(\boldsymbol{x})$ is the same for every $\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}}$. Therefore we have the optimal solution to Problem (3):

$$q_{\mathcal{S}}^{\star}(\boldsymbol{x}) = \begin{cases} p(\boldsymbol{x}) + \frac{1-C}{|\mathcal{X}_{\mathcal{S}}|}, & \boldsymbol{x} \in \mathcal{X}_{\mathcal{S}} \\ 0, & \boldsymbol{x} \notin \mathcal{X}_{\mathcal{S}} \end{cases}$$

Computational complexity. The time we need to solve Problem (3) is $O(|\mathcal{X}_{\mathcal{S}}|)$, i.e. the time to compute C. However, to compute the norm $\|q(\cdot) - p(\cdot)\|_2^2$ we still need $O(|\mathcal{X}|)$ time, as $p(\boldsymbol{x})$ is not necessarily zero at any $\boldsymbol{x} \in \mathcal{X}$. As a result, we need $O(n^k(|\mathcal{X}| + |\mathcal{X}_{\mathcal{S}}|))$ time to enumerate for an optimal solution of the ℓ_2-norm projection. If we consider the integer lattice $\mathcal{X}$, as stated in the problem setting, then $|\mathcal{X}| = m^n$ and $|\mathcal{X}_{\mathcal{S}}| = m^k$, rendering the time complexity $O(n^k m^n)$. However, Algorithm 2 has much lower time complexity. In each iteration, the greedy method selects an element to put into $\mathcal{S}$ that maximize the gain, which requires k iterations. It need not to consider the exact ℓ_2-norm $\|q(\cdot) - p(\cdot)\|_2^2$ in each iteration, only the increment for each e from n options. To compute the increment, no more than $|\mathcal{X}_{\mathcal{S}}|$ terms are added, which requires compute of $O(|\mathcal{X}_{\mathcal{S}}|)$ time complexity. All together, the greedy method requires $O(k|\mathcal{X}_{\mathcal{S}}|)$ time to operate, or $O(nkm^k)$ in our integer lattice setting, which is far less that the enumeration method's $O(n^k m^n)$.

3.4 When Greedy is Good

We have shown in the proof of Theorem 2 that there always exist extreme examples that are hard to solve. Thus, in the most general sense, and without further assumptions, one can find pathological cases which make the problem hard. However, we find that the greedy approach works well empirically. In this section, we consider sufficient conditions for tractability of the problem. Our conditions boil down to structural assumptions on $F[\cdot]$ which match standard assumptions in the literature.

To build further intuition, consider line 4 in Algorithm 1, where the parameter passed to the greedy method is $q(\cdot) = p(\cdot) - \mu \frac{\delta F}{\delta p}(\cdot)$, and $p(\cdot)$ is already a k-sparse distribution. Denote the support of $p(\cdot)$ as $\mathcal{S}$; we can see that $|\mathcal{S}| \leq k$. Therefore, that $q(\cdot)$ is close to k-sparse when the step size μ is small. Thus, while the general problem (2) may be a lot harder, there is reason to conjecture that under certain conditions, a simple greedy algorithm performs well. Next, we state these assumptions formally.

Assumption 1 (**Strong Convexity/Smoothness**). *The objective $F[\cdot]$ satisfies Strong Convexity/Smoothness with respect to α and β if:*

$$\frac{\alpha}{2}\|p_1(\cdot) - p_2(\cdot)\|_2^2 \leq F[p_1(\cdot)] - F[p_2(\cdot)] - \left\langle \frac{\delta F}{\delta p_1}(\cdot), p_2(\cdot) - p_1(\cdot) \right\rangle \leq \frac{\beta}{2}\|p_1(\cdot) - p_2(\cdot)\|_2^2$$

For the sake of simplicity in exposition, we have assumed strong convexity to hold over the entire domain (which can be a restrictive assumption). As will be clear from the proof analysis, this assumption can easily be tightened to a restricted strong convexity assumption; see, e.g., [30]. This detail is left for a longer version of this manuscript.

Assumption 2 (**Lipschitz Condition**). *The functional $F : \mathcal{P} \to \mathbb{R}$ satisfies the Lipschitz condition with respect to L, in k-sparse domain $\mathcal{D}_k$ is*

$$|F[p_1(\cdot)] - F[p_2(\cdot)]| \leq L\|p_1(\cdot) - p_2(\cdot)\|_2$$

This assumption implies that

$$\left\| \frac{\delta F}{\delta p}(\cdot) \right\|_2 \leq L.$$

Using the strong convexity, smoothness, and Lipschitz assumptions, we are able to provide analysis for when greedy works well. This is encapsulated in Theorem 3.

Theorem 3. *Given n-dimensional function $q(\cdot) = p(\cdot) - \mu \frac{\delta F}{\delta p}(\cdot)$, where $p(\cdot)$ is an n-dimensional k-sparse distribution and supp$(p(\cdot)) = \mathcal{S}'$, Algorithm 2 finds the optimal projection to domain $\mathcal{P}_{\mathcal{S}'}$ if $F[\cdot]$ satisfies Assumption 2, μ is sufficiently small and there are enough positions $\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}'}$ where $p(\boldsymbol{x}) > 0$, i.e., satisfies inequality* (6) *and inequality* (9).

3.5 Convergence Analysis

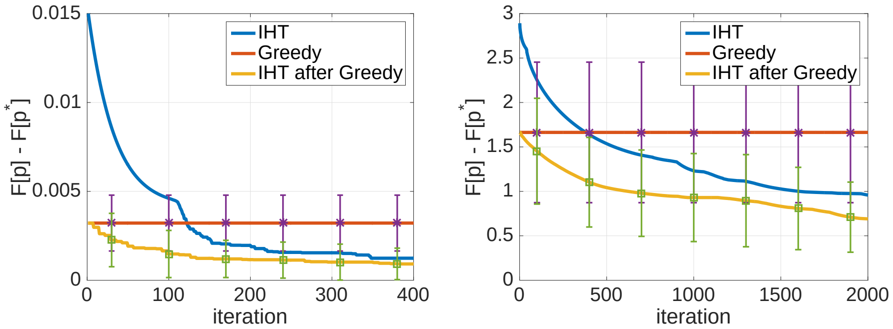
Next, we analyze the convergence of the overall Algorithm 1 with greedy projections. While Theorem 3 provides sufficient conditions for exact projection using the greedy approach, in practice due to computational precision issues and/or violation of the stated assumptions, the solution may not provide an exact projection. Thus, it is prudent to assume that the inner projection subproblem is solved within some approximation as quantified in the following.

Definition 5. ***Approximate ℓ_2-norm projection***. *We define $\widehat{\Pi}_{\mathcal{D}_k}(\cdot)$ as the approximate projection onto sparsity domain and distribution space, with approximation parameter, ϕ, as:*

$$\left\| p(\cdot) - \widehat{\Pi}_{\mathcal{D}_k}(p(\cdot)) \right\|_2^2 \leq (1+\phi) \left\| p(\cdot) - \Pi_{\mathcal{D}_k}(p(\cdot)) \right\|_2^2$$

Next, we present our main convergence theorem.

Theorem 4. *Suppose F satisfies assumptions 1 and 2. Furthermore, assume that the projection step in Algorithm 1 is solved ϕ-approximately. Let the step size $\mu = 1/\beta$, and $\|p_0(\cdot) - p^\star(\cdot)\|_2 \leq L/(2\alpha)$. Then if $\frac{\beta}{\alpha} \in (2 - \frac{1}{1+\phi}, 2)$, IHT (Algorithm 1) with $T \geq \log_\eta \frac{\epsilon}{F[p_0(\cdot)] - F[p^\star(\cdot)] - c}$ iterations achieves $F[p_T(\cdot)] \leq F[p^\star(\cdot)] + c + \epsilon$, where $\eta = 1 - (1+\phi)(2 - \beta/\alpha)$ and $c = \frac{\left(\phi/(2\beta) + (1+\phi)(\beta-\alpha)/(2\alpha^2)\right)L^2}{(1+\phi)(2-\beta/\alpha)}$.*



(a) Normalized ℓ_2-norm Minimization (b) Normalized KL Minimization

Figure 2: Simulated Experiments

4 Experiments

We evaluate our algorithm on different convex objectives, namely, ℓ_2-norm distance and KL divergence. As mentioned before, there are no theoretically guaranteed algorithms for ℓ_2-norm distance minimization under sparsity constraint. To investigate optimality of the algorithms, we consider simulated experiments of sufficiently small size that the global optimal can be exhaustively enumerated.

Algorithm 3 Greedy Selection

```
Input: $F[\cdot] : \mathcal{P} \to \mathbb{R}$, $k \in \mathbb{Z}_+$. Output: $p_T \in \mathcal{D}_k$
$\mathcal{S} := \emptyset$
while $|\mathcal{S}| < k$ do
    $j \in \arg\min_{i \in [n] \setminus \mathcal{S}} \{\min_{p \in \mathcal{P}_{\mathcal{S} \cup i}} F[p(\cdot)]\}$
    $\mathcal{S} := \mathcal{S} \cup j$
end while
return $\arg\min_{p \in \mathcal{P}_\mathcal{S}} F[p(\cdot)]$
```

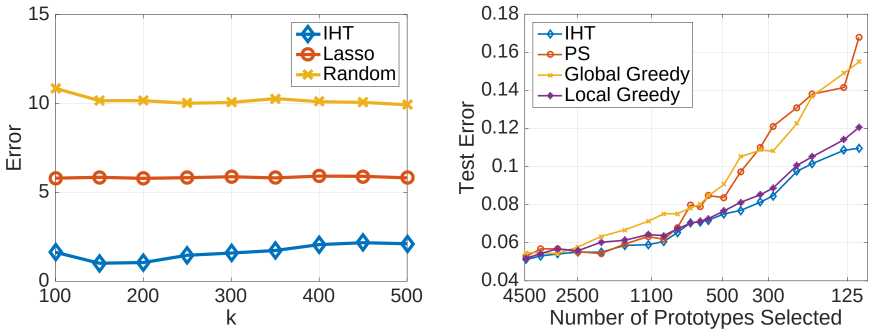
IHT implementation details. For IHT, the step size is chosen by a simple strategy: given an initial step size, we double the step size when IHT is trapped in local optima, and return to the initial step size after escaping. We return the algorithm along the entire solution path.

Baseline: Forward Greedy Selection. Unfortunately, we are unaware of optimization algorithms for sparse probability estimation with general losses. As as a simple baseline, we consider greedy selection wrt. the objective. This is equivalent to Algorithm 3. For certain special cases e.g. KL objective, Algorithm 3 can be applied efficiently and is effective in practice [5].

4.1 Simulated Data

We set dimension $n = 15$, number of entries $m = 2$, sparsity level $k = 7$. That is, $\mathcal{X} = \{0,1\}^{15}$ is a 15-dimensional binary vector space, with cardinality $|\mathcal{X}| = 2^{15} = 32768$. The distribution $p : \mathcal{X} \to [0,1]$ satisfies $\sum_{\boldsymbol{x} \in \mathcal{X}} p(\boldsymbol{x}) = 1$. The sparsity constraint is designed to fix a support $\mathcal{S} : |\mathcal{S}| \leq 7$, such that for any $\boldsymbol{x} : p(\boldsymbol{x}) > 0$ has $\operatorname{supp}(\boldsymbol{x}) = \mathcal{S}$. Thus, the optimal solution is requires enumerating $\binom{15}{7} = 6435$ possible supports.

The ℓ_2-norm minimization objective is $F[p(\cdot)] = \|p(\cdot) - q(\cdot)\|_2^2$ where $q(\cdot)$ is a distribution generated by randomly choosing 50 positions $x_1 \cdots x_{50} \in \mathcal{X}$ to assign random real numbers



(a) (b)

Figure 3: (a) Compression / Compressed sensing. Test Error at varying sparsity k. (b) Dataset Compression. Test Classification error of prototype nearest neighbor classifier

$c_1 \cdots c_{50} : \sum_{i=1}^{50} c_i = 1$ and the other positions are assigned to 0, i.e., $q(x_i) = c_i$ for $i \in [50]$, and $q(\boldsymbol{x}) = 0$ otherwise. Initial step size $\mu = 0.008$. Results are shown in Figure 2 (a). For the KL divergence objective, it is $F[p(\cdot)] = KL(p(\cdot)||q(\cdot)) = \sum_{\boldsymbol{x} \in \mathcal{X}} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}$, where $q(\cdot)$ is a random distribution generated similar to the $q(\cdot)$ in ℓ_2-norm objective. The only difference is that $q(\boldsymbol{x})$ can not be zero as it would render the KL undefined. For simulated experiments, we use the optimum to normalize the objective function as $\tilde{F}[p] = F[p] - F[p^\star]$, so that at the optimum $\tilde{F}[p^\star] = 0$.

Three algorithms are compared in each experiment, i.e., IHT, Greedy and IHT after Greedy. While IHT starts randomly, IHT after Greedy is initialized by the result of Greedy. In each run, the distribution $q(\cdot)$ and the starting distribution for IHT $p_0(\cdot)$ are randomly generated. Each of the experiments are run 20 times. Results are presented in showing the mean and standard deviation of Greedy and IHT after Greedy. The standard deviation of IHT is similar to that of IHT after Greedy.

We use the ℓ_2-norm greedy projection in IHT in both experiments. Interestingly, this not only outperforms the ℓ_2-norm greedy projection itself (Figure 2 (a)), but also outperforms Greedy on the KL objective (Figure 2 (b)), where [5] suggests provably good performance. In particularly, while the performance of Greedy can fluctuate severely, IHT (after Greedy) is stable in obtaining good results. Note that low variance is especially desirable when the algorithm is only applied a few times to save computation, as in large discrete optimization problems.

4.2 Benchmark Data

Distribution Compression / Compressed sensing. We apply our IHT to the task of expectation-preserving distribution compression, useful for efficiently storing large probability tables. Given a distribution $p(\cdot)$, our goal is to construct a sparse approximation $q(\cdot)$, such that $q(\cdot)$ approximately preserves expectations with respect to $p(\cdot)$. Interestingly, this model compression problem is equivalent to compressed sensing, but with the distributional constraints. Specifically, our goal is to find $\boldsymbol{q}$ which minimizes $||A\boldsymbol{q} - A\boldsymbol{p}||_2^2$ subject to a k-sparsity constraint on $\boldsymbol{q}$. The model is evaluated with respect to moment reconstruction $||B\boldsymbol{q} - B\boldsymbol{p}||_2^2$ for a new "sensing" matrix B. Our experiments use real data from the Texas hospital discharge public use dataset. IHT is compared to post-precessed Lasso and Random. Lasso ignores the simplex constraints during optimization, then projects the results to the simplex, while Random is a naïve baseline of random distributions. Figure 3(a) shows that IHT significantly outperforms baselines. Additional details are provided in Appendix H due to limited space.

Dataset compression. We study representative prototype selection for the Digits data [31]. Prototypes are representative examples chosen from the data in order to achieve dataset compression. Our optimization objective is the Maximum Mean Discrepancy (MMD) between the discrete data distribution and the sparse data distribution representing the selected samples. We evaluate performance using the prototype nearest neighbor classification error on a test dataset. We compare two forward selection greedy variants (Local Greedy and Global Greedy) proposed by [32] and the means algorithm (labeled as PS) proposed by [33], both state of the art. The results are presented in Figure 3(b) showing that IHT outperforms all baselines. Additional experimental details are provided in Appendix H due to limited space.

5 Conclusion and Future Work

In this work, we proposed the use of IHT for learning discrete sparse distributions. We study several theoretical properties of the algorithm from an optimization viewpoint, and propose practical solutions to solve otherwise hard problems. There are several possible future directions of research. We have analyzed discrete distributions with sparsity constraints. The obvious extensions are to the space of continuous measures and structured sparsity constraints. Is there a bigger class of constraints for which the a tractable projection algorithm exists? Can we improve the sufficient conditions under which projections are provably close to the optimum projection? Finally, more in-depth empirical studies compared to other state of the art algorithms should be very interesting and useful to the community.

References

[1] T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

[2] Hemant Ishwaran and J. Sunil Rao. Spike and slab variable selection: Frequentist and bayesian strategies. *Ann. Statist.*, 33(2):730–773, 04 2005.

[3] Edward I. George and Robert E. McCulloch. Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

[4] Trevor Park and George Casella. The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

[5] Oluwasanmi O Koyejo, Rajiv Khanna, Joydeep Ghosh, and Russell Poldrack. On prior distributions and approximate inference for structured variables. In *Advances in Neural Information Processing Systems*, pages 676–684, 2014.

[6] Rajiv Khanna, Joydeep Ghosh, Russell Poldrack, and Oluwasanmi Koyejo. Sparse submodular probabilistic pca. In *Artificial Intelligence and Statistics*, pages 453–461, 2015.

[7] Rajiv Khanna, Joydeep Ghosh, Rusell Poldrack, and Oluwasanmi Koyejo. Information Projection and Approximate Inference for Structured Sparse Variables. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1358–1366, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

[8] CARLOS M. CARVALHO, NICHOLAS G. POLSON, and JAMES G. SCOTT. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

[9] Andre Wibisono. Sampling as optimization in the space of measures: The langevin dynamics as a composite optimization problem. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 2093–3027. PMLR, 06–09 Jul 2018.

[10] Francesco Locatello, Rajiv Khanna, Joydeep Ghosh, and Gunnar Ratsch. Boosting variational inference: an optimization perspective. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 464–472, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR.

[11] Arnak S.Dalalyan and Avetik Karagulyan. User-friendly guarantees for the langevin monte carlo with inaccurate gradient. *Stochastic Processes and their Applications*, 2019.

[12] Ferenc Huszár and David Duvenaud. Optimally-weighted herding is bayesian quadrature. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, UAI'12, pages 377–386, 2012.

[13] Francesco Locatello, Gideon Dresdner, Rajiv Khanna, Isabel Valera, and Gunnar Raetsch. Boosting black box variational inference. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 3401–3411. Curran Associates, Inc., 2018.

[14] Thomas Blumensath and Mike E. Davies. Iterative hard thresholding for compressed sensing. *CoRR*, 2008.

[15] 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)*, pages 353–356. IEEE, 2011.

[16] 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.

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

[18] Joel A Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information theory*, 50(10):2231–2242, 2004.

[19] 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, pages 685–693, 2014.

[20] Rajiv Khanna and Anastasios Kyrillidis. Iht dies hard: Provable accelerated iterative hard thresholding. *AISTATS*, 12 2018.

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

[22] Robert Tibshirani, Martin Wainwright, and Trevor Hastie. *Statistical learning with sparsity: the lasso and generalizations*. Chapman and Hall/CRC, 2015.

[23] Suvrit Sra, Sebastian Nowozin, and Stephen J Wright. *Optimization for machine learning*. Mit Press, 2012.

[24] Junzhou Huang, Tong Zhang, and Dimitris Metaxas. Learning with structured sparsity. *Journal of Machine Learning Research*, 12(Nov):3371–3412, 2011.

[25] Eberhard Engel and Reiner M Dreizler. *Density functional theory*. Springer.

[26] Jon Kleinberg and Eva Tardos. *Algorithm design*. Pearson Education India, 2006.

[27] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279. ACM, 2008.

[28] Anastasios Kyrillidis and Volkan Cevher. Matrix recipes for hard thresholding methods. *Journal of mathematical imaging and vision*, 48(2):235–265, 2014.

[29] Anastasios Kyrillidis, Stephen Becker, Volkan Cevher, and Christoph Koch. Sparse projections onto the simplex. In *International Conference on Machine Learning*, pages 235–243, 2013.

[30] Alekh Agarwal, Sahand Negahban, and Martin J Wainwright. Fast global convergence rates of gradient methods for high-dimensional statistical recovery. In *Advances in Neural Information Processing Systems*, pages 37–45, 2010.

[31] J. J. Hull. A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(5):550–554, May 1994.

[32] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2280–2288. Curran Associates, Inc., 2016.

[33] Jacob Bien and Robert Tibshirani. Prototype selection for interpretable classification. *Ann. Appl. Stat.*, 5(4):2403–2424, 12 2011.

A Vector-Sparsity for Distributions

While our framework is developed for sparsity along the dimensions of a multivariate discrete distribution, it is easily extended to alternative notions of sparsity. One common setting is where we are interested in sparsity of the distribution $p(\cdot)$ when represented as a vector $\mathbf{p}$ e.g. sparsifying the number of valid states of a univariate distribution such as a histogram. In our setting, this can be solved by constructing a binary vector $Z \in \mathcal{Z} = \{0,1\}^{|\mathcal{X}|}$, where each dimension of Z indexes one of the possible states of X. As each state of X is associated with an element of the vector $\mathbf{p}$, state restrictions imply vector sparsity of $\mathbf{p}$. For example $\mathbf{z} = [1, 1, \ldots, 1, 0]$ implies that X can take all states apart from the last one, $\mathbf{z} = [1, 1, 0, \ldots, 0]$ implies that only the first two states are valid, and so on. Setting $\mathbb{P}(Z = \mathbf{z}) \propto \mathbb{P}(X \in \{\text{states indexed by } \mathbf{z}\}) \propto \mathbf{z}^\top \mathbf{p}$ completes the transformation. In summary, dimension-wise sparsity of $\mathbf{z}$ corresponds to restrictons on the support of X, which equivalently corresponds to sparsification of the vector probability $\mathbf{p}$.

B Connection between Variational Derivative and Gradient

As random variable has discrete space $\mathcal{X}$, we can use a vector to store probability of every $\boldsymbol{x} \in \mathcal{X}$ of a distribution $q(\cdot) : \mathcal{X} \to [0, 1]$. That is, the vector serves as an oracle of $q(\cdot)$. We denote the vector as $\widehat{\boldsymbol{q}} \in [0, 1]^{|\mathcal{X}|}$ to distinguish it from the original $q(\cdot) \in \mathcal{P}$.

We define a bijection map $\Phi(\cdot) : \mathcal{X} \to [|\mathcal{X}|]$, and let $\forall \boldsymbol{x} \in \mathcal{X}$, we have $q(\boldsymbol{x}) = \widehat{q}_{\Phi(\boldsymbol{x})}$, where $\widehat{q}_{\Phi(\boldsymbol{x})}$ is the $\Phi(\boldsymbol{x})^{th}$ entry of vector $\widehat{\boldsymbol{q}}$. In this case, we may consider $\widehat{F}(\widehat{\boldsymbol{q}})$ as a function in vector space, *i.e.*, $\widehat{F}(\cdot) : [0, 1]^{|\mathcal{X}|} \to \mathbb{R}$, and $F[q] = \widehat{F}(\widehat{\boldsymbol{q}})$. That is, we have re-represent the original functional $F : \mathcal{P} \to \mathbb{R}$ as a function $\widehat{F} : [0, 1]^{|\mathcal{X}|} \to \mathbb{R}$. The function $\widehat{F}(\cdot)$ naturally has its gradient.

Definition 6. ***Gradient of*** $\widehat{F}(\cdot)$. *The gradient of* $\widehat{F}(\cdot) : [0, 1]^{|\mathcal{X}|} \to \mathbb{R}$ *is*

$$\nabla \widehat{F}(\widehat{\boldsymbol{q}}) = \left[\frac{\partial \widehat{F}}{\partial \widehat{q}_1}, \cdots, \frac{\partial \widehat{F}}{\partial \widehat{q}_{|\mathcal{X}|}}\right]^\top$$

We next show that no matter which bijection map is chosen, the gradient of $\widehat{F}(\cdot)$, *i.e.*, Definition 6. and the variational derivative of $F[\cdot]$, *i.e.*, Definition 3, are equivalent.

Theorem 5. *In the case that $\mathcal{X}$ is discrete, definition 3 is equivalent to definition 6 given any bijection* $\Phi(\cdot) : \mathcal{X} \to [|\mathcal{X}|]$, i.e.,

$$\frac{\delta F}{\delta q}(\boldsymbol{x}) = \nabla \widehat{F}(\widehat{\boldsymbol{q}})_{\Phi(\boldsymbol{x})}$$

for any $\boldsymbol{x} \in \mathcal{X}$, where $\nabla \widehat{F}(\widehat{\boldsymbol{q}})_{\Phi(\boldsymbol{x})}$ is the $\Phi(\boldsymbol{x})^{th}$ entry of gradient vector $\nabla \widehat{F}(\widehat{\boldsymbol{q}})$.

Proof. First we define an operator $\texttt{vec}(\cdot)$ from function space over $\mathcal{X}$ to vector space $\mathbb{R}^{|\mathcal{X}|}$, so that for a function $f(\cdot) : \mathcal{X} \to \mathbb{R}$ and every $\boldsymbol{x} \in \mathcal{X}$, we have $f(\boldsymbol{x}) = \texttt{vec}(f)_{\Phi(\boldsymbol{x})}$. That is, by using $\texttt{vec}(\cdot)$, we store every information of $f(\cdot)$ into a vector. Therefore, by substituting $\frac{\delta F}{\delta q}(\boldsymbol{x})$ by $\nabla \widehat{F}(\widehat{\boldsymbol{q}})_{\Phi(\boldsymbol{x})}$ in definition 3, we have:

$$\sum_{\boldsymbol{x} \in \mathcal{X}} \nabla \widehat{F}(\widehat{\boldsymbol{q}})_{\Phi(\boldsymbol{x})} \phi = \texttt{vec}(\phi)^\top \nabla \widehat{F}(\widehat{\boldsymbol{q}}) = \lim_{\epsilon \to 0} \frac{\widehat{F}(\widehat{\boldsymbol{q}} + \epsilon \texttt{vec}(\phi)) - \widehat{F}(\widehat{\boldsymbol{q}})}{\epsilon}$$

$$= \left[\frac{d}{d\epsilon} \hat{F}(\hat{\boldsymbol{q}} + \epsilon \texttt{vec}(\phi))\right]_{\epsilon=0} = \left[\frac{d}{d\epsilon} \hat{F}(\texttt{vec}(q + \epsilon\phi))\right]_{\epsilon=0} = \left[\frac{dF[q + \epsilon\phi]}{d\epsilon}\right]_{\epsilon=0}$$

This shows that $\frac{\delta F}{\delta q} = \texttt{vec}^{-1}(\nabla \hat{F}(\hat{\boldsymbol{q}})_{\Phi(\boldsymbol{x})})$, *i.e.*, $\frac{\delta F}{\delta q}(\boldsymbol{x}) = \nabla \widehat{F}(\widehat{\boldsymbol{q}})_{\Phi(\boldsymbol{x})}$ for any $\boldsymbol{x} \in \mathcal{X}$, where the inverse of $\texttt{vec}(\cdot)$ exists because that the $\Phi(\cdot)$ is bijection. □

Though they are equivalent in discrete settings, we can see that definition 3 is more general than definition 6, as the variational derivative can be easily extended to continuous $\mathcal{X}$. Even when $\mathcal{X}$ is discrete, it can also be used when q is given by a function, with no need to store everything in a vector.

C Discussion on Other Projection Heuristics

One may ponder how may other heuristics perform when dealing with the ℓ_2-norm sparse projection. For example, a two-stage thresholding approach, i.e. $i)$ running gradient descent to convergence, and then $ii)$ projection. However, it is known to be sub-optimal even for simple problems, such as least-squares with sparsity constraints. In fact, the results when using such approach on ℓ_2-norm minimization are the same as the Greedy baseline: It $i)$ converges to the global optimum $q(\cdot)$, and then $ii)$ use greedy projection to try to minimize $F[p(\cdot)] = \|p(\cdot) - q(\cdot)\|_2^2$ subject to sparsity constraint, which has been shown to be inferior than IHT (subsection 4.1).

One may also come up with the idea of choosing the k "heaviest" coordinates as support. However, in the general case, taking the k-heaviest coordinates of q (without assuming any structure) would result into a non-valid putative solution; recall, by definition of the discrete setting, we have n coordinates, each of which takes m points, leading to a m^n sample space. Simply taking the k-heaviest coordinates of that long vector would result into an intermediate representation of non-zero positions that does not correspond to a probability distribution. A variation of this approach is fine for the "vector-sparsity" special case.

D Proof of Theorem 1

Here, we show that the subset selection problem can be reduced to the sparse l_2 distribution l_2-norm projection problem (2). Let us first define the subset sum problem, or SSP.

Definition 7 (Subset Sum Problem [26])**.** *Given a ground set of integers, $\mathcal{G} \subset \{\mathbb{Z}\}^n$, in the Subset Sum Problem we look for a non-empty subset $\mathcal{S} \subseteq \mathcal{G}$, such that the sum of all elements in $\mathcal{S}$ is zero. This is an NP-complete problem.*

Consider a Subset Sum Problem instance with a ground set $\mathcal{G}$, where $|\mathcal{G}| = n$. Let us denote its elements as $e_1, \ldots, e_n$. We reformulate the sparse distribution ℓ_2-norm projection problem as follows. Let $\mathcal{X}$ be n-dimensional binary space, *i.e.*, $\mathcal{X} = \{0,1\}^n$. For $\boldsymbol{x} \in \mathcal{X}$, let its positive positions denote a subset, *i.e.*, $\mathcal{G}_{\boldsymbol{x}} = \{e_i \mid \boldsymbol{x}_i = 1\}$. Define an n-dimensional function $q_k : \mathcal{X} \to \mathbb{R}$ as

$$q_k(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \sum_{e \in \mathcal{G}_{\boldsymbol{x}}} e = 0 \quad \text{and} \quad |\mathcal{G}_{\boldsymbol{x}}| = k, \\ 0, & \text{otherwise,} \end{cases}$$

where k is a parameter.

Then, we try to find its projection to $\mathcal{D}_k$ from $k = 1$ to $k = n$. Denote $\widehat{p}_k(\cdot)$ as the optimal ℓ_2-norm projection of $q_k(\cdot)$ to k-sparse distribution set $\mathcal{D}_k$. We can see that, if there is no subset $\mathcal{G}_{\boldsymbol{x}}$ with size k that sum up to zero, then $q_k(\cdot)$ is zero everywhere. Denote the support of the optimal projection $\widehat{p}_k(\cdot)$ as $\mathcal{S}^\star$, and we can see that $\widehat{p}_k(\boldsymbol{x}) = \frac{1}{2^k}$ for every supp$(\boldsymbol{x}) \subseteq \mathcal{S}^\star$. That is, $\widehat{p}_k(\mathbf{0}) = \frac{1}{2^k}$, since $supp(\mathbf{0}) = \emptyset \subseteq \mathcal{S}^\star$.

If there exist a subset $\mathcal{G}_{\boldsymbol{x}'}$ with size k that sum up to zero, we can see that $\widehat{p}_k(\boldsymbol{x}) = 1$ when $\boldsymbol{x} = \boldsymbol{x}'$ and $\widehat{p}_k(\boldsymbol{x}) = 0$ elsewhere. Therefore, noting that $|supp(\boldsymbol{x}')| = k$, we can check whether $\widehat{p}_k$ –*i.e.*, the optimal ℓ_2-norm projection of $q_k(\cdot)$ to $\mathcal{D}_k$– has $\widehat{p}_k(\mathbf{0}) = \frac{1}{2^k}$, to know that whether there exist a subset $\mathcal{G}_{\boldsymbol{x}} \subseteq \mathcal{G}$ with size k summing up to zero.

If there exist a polynomial algorithm solving the projection problem in $O(\text{poly}(n))$, then we can run it $O(n)$ times to try from $k = 1$ to $k = n$, to solve the Subset Sum Problem. Since the Subset Sum Problem is NP-hard, hence the NP-hardness of the sparse distribution ℓ_2-norm projection problem.

E Proof of Theorem 2

We prove the theorem by showing that, for any algorithm we can design an example where the algorithm fails. Note that in Algorithm 1 the input of projection step is not necessarily a distribution. That is, we have to consider the input of the projection problem (2) a general function. Let $\mathcal{X}$ be n-dimensional binary space, *i.e.*, $\mathcal{X} = \{0,1\}^n$. Denote an always-zero function $q_0 : \mathcal{X} \to 0$. Given an deterministic algorithm f, it takes in a function $q : \mathcal{X} \to \mathbb{R}$ and output a distribution $\widehat{p}(\cdot)$ with support $\mathcal{S}$, where $|\mathcal{S}| = k$. Assume the algorithm f evaluates $T = O(poly(n))$ positions, denoting as $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T$. Note that T can be much less than $\binom{n}{k}$, as $\binom{n}{k}$ cannot be upper bound by any n^c where c is a constant. Therefore, there exist an $\boldsymbol{x}^\star$ as

$$\boldsymbol{x}^\star \in \mathcal{X} \backslash \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\} \quad \text{s.t.} \quad \text{supp}(\boldsymbol{x}) \neq \mathcal{S} \quad \textit{and} \quad |\text{supp}(\boldsymbol{x})| = k$$

Now we construct an n-dimensional function $q : \mathcal{X} \to \mathbb{R}$ as

$$q(\boldsymbol{x}) = \begin{cases} 1+\delta, & \text{if} \quad \boldsymbol{x} = \boldsymbol{x}^\star \\ 0, & \text{otherwise} \end{cases},$$

where $\delta > 0$. We input the constructed q to the deterministic algorithm f. Note that the value of positions it evaluates do not have any differences compared to those when q_0 is inputed, *i.e.*, $q(\boldsymbol{x}_i) = q_0(\boldsymbol{x}_i) = 0$ for every $i \in [T]$. As f is deterministic, the output solution is still $\widehat{p}$ with support $\mathcal{S}$. As a result, $q(\boldsymbol{x}) = 0$ for every $\boldsymbol{x} \in \mathcal{X}_\mathcal{S}$. Denoting $\widehat{p}_\mathcal{S}$ as the optimal ℓ_2-norm projection of q to $\mathcal{P}_\mathcal{S}$, we have $\|q(\cdot) - \widehat{p}(\cdot)\|_2^2 \geq \|q(\cdot) - \widehat{p}_\mathcal{S}\|_2^2 = 1/|\mathcal{X}_\mathcal{S}| + (1+\delta)^2$.

Noting that the optimal projection is

$$p^\star(\boldsymbol{x}) = \begin{cases} 1, & \text{if} \quad \boldsymbol{x} = \boldsymbol{x}^\star \\ 0, & \text{otherwise} \end{cases},$$

we can see the optimal ℓ_2-norm distance is $\|q(\cdot) - p^\star(\cdot)\|_2^2 = \delta^2$. Therefore, the approximation rate of algorithm f on this input q is

$$\varphi = \frac{\|q - \widehat{p}\|_2^2}{\|q - p^\star\|_2^2} - 1 \geq \frac{\frac{1}{|\mathcal{X}_\mathcal{S}|} + (1+\delta)^2}{\delta^2} - 1 \geq \frac{\frac{1}{|\mathcal{X}_\mathcal{S}|} + 1}{\delta^2} - 1$$

As δ can be arbitrarily close to 0, the approximation ratio φ can not be upper bounded.

F Proof of Theorem 3

Proof. First, we quantify the influence of the gradient step. For any support $\mathcal{S} \subset [n]$, let $\hat{q}_\mathcal{S}$ be the optimal projection of $q = p - \mu\frac{\delta F}{\delta p}$ to sparsity domain $\mathcal{P}_\mathcal{S}$, and let $\hat{p}_\mathcal{S}$ be the optimal projection of p to sparsity domain $\mathcal{P}_\mathcal{S}$. Then, we have

$$\|\hat{q}_\mathcal{S} - q\|_2 = \left\|\hat{q}_\mathcal{S} - p + \mu\frac{\delta F}{\delta p}\right\|_2 \geq \|\hat{q}_\mathcal{S} - p\|_2 - \mu L \geq \|\hat{p}_\mathcal{S} - p\|_2 - \mu L$$

Its upper bound is

$$\|\hat{q}_\mathcal{S} - q\|_2 \leq \|\hat{p}_\mathcal{S} - q\|_2 = \left\|\hat{p}_\mathcal{S} - p + \mu\frac{\delta F}{\delta p}\right\|_2 \leq \|\hat{p}_\mathcal{S} - p\|_2 + \mu L$$

That is,

$$\|\hat{p}_\mathcal{S} - p\|_2 + \mu L \geq \|\hat{q}_\mathcal{S} - q\|_2 \geq \|\hat{p}_\mathcal{S} - p\|_2 - \mu L \tag{5}$$

Nest, consider a support $\mathcal{S} \subset \mathcal{S}'$, where $\mathcal{S}' = supp(p)$. We use the greedy procedure to add one element $e \in [n] \backslash \mathcal{S}$ to $\mathcal{S}$. It is to find

$$e \in \arg\min_{i \in [n] \backslash \mathcal{S}} \|\hat{q}_{\mathcal{S} \cup i} - q\|_2$$

Define θ as a parameter describing how much better if we choose support $\mathcal{S} \subset \mathcal{S}'$ to project than choosing other supports.

$$\theta = \min_{\mathcal{S}: \mathcal{S} \subset \mathcal{S}'} \left(\min_{i \in [n] \backslash \mathcal{S}', j \in \mathcal{S}' \backslash \mathcal{S}} \|\hat{q}_{\mathcal{S} \cup i} - q\|_2 - \|\hat{q}_{\mathcal{S} \cup j} - q\|_2 \right)$$

As we can see, the greater θ is, the more possible for the greedy method to finally find $\mathcal{S}'$. Moreover, if $\theta > 0$, then the greedy procedure finds exactly $\mathcal{S}'$. By using inequality (5), we have

$$\theta > \min_{\mathcal{S}: \mathcal{S} \subset \mathcal{S}'} \left(\min_{i \in [n] \backslash \mathcal{S}', j \in \mathcal{S}' \backslash \mathcal{S}} \|\hat{p}_{\mathcal{S} \cup i} - p\|_2 - \|\hat{p}_{\mathcal{S} \cup j} - p\|_2 \right) - 2\mu L$$

Therefore, if we have

$$2\mu L < \min_{\mathcal{S}: \mathcal{S} \subset \mathcal{S}'} \left(\min_{i \in [n] \backslash \mathcal{S}', j \in \mathcal{S}' \backslash \mathcal{S}} \|\hat{p}_{\mathcal{S} \cup i} - p\|_2 - \|\hat{p}_{\mathcal{S} \cup j} - p\|_2 \right), \tag{6}$$

we can guarantee $\theta > 0$, which means the greedy method finds exactly $\mathcal{S}'$.

Next, we analyze when is inequality (6) achievable for enough small step size $\mu > 0$, *i.e.*, $\|\hat{p}_{\mathcal{S}\cup i} - p\|_2 - \|\hat{p}_{\mathcal{S}\cup j} - p\|_2 > 0$ in inequality (6).

First, let us calculate $\|\hat{p}_{\mathcal{S}\cup i} - p\|_2^2$ for any $i \in [n]\backslash\mathcal{S}'$.

$$\begin{aligned}
\|\hat{p}_{\mathcal{S}\cup i} - p\|_2^2 &= \sum_{supp(\boldsymbol{x})\le|\mathcal{S}\cup i|} (\hat{p}_{\mathcal{S}\cup i}(\boldsymbol{x}) - p(\boldsymbol{x}))^2 + \sum_{supp(\boldsymbol{x})>|\mathcal{S}\cup i|} (\hat{p}_{\mathcal{S}\cup i}(\boldsymbol{x}) - p(\boldsymbol{x}))^2 \\
&= \sum_{supp(\boldsymbol{x})\le|\mathcal{S}\cup i|} (\hat{p}_{\mathcal{S}\cup i}(\boldsymbol{x}) - p(\boldsymbol{x}))^2 + \sum_{supp(\boldsymbol{x})>|\mathcal{S}\cup i|} p(\boldsymbol{x})^2 \\
&= \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup i}} (\hat{p}_{\mathcal{S}\cup i}(\boldsymbol{x}) - p(\boldsymbol{x}))^2 + \sum_{supp(\boldsymbol{x})\le|\mathcal{S}\cup i|, \boldsymbol{x}\notin\mathcal{X}_{\mathcal{S}\cup i}} (\hat{p}_{\mathcal{S}\cup i}(\boldsymbol{x}) - p(\boldsymbol{x}))^2 \\
&\quad + \sum_{supp(\boldsymbol{x})>|\mathcal{S}\cup i|} p(\boldsymbol{x})^2 \\
&= \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup i}} (\hat{p}_{\mathcal{S}\cup i}(\boldsymbol{x}) - p(\boldsymbol{x}))^2 + \sum_{supp(\boldsymbol{x})\le|\mathcal{S}\cup i|, \boldsymbol{x}\notin\mathcal{X}_{\mathcal{S}\cup i}} p(\boldsymbol{x})^2 + \sum_{supp(\boldsymbol{x})>|\mathcal{S}\cup i|} p(\boldsymbol{x})^2 \\
&= \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup i}} (\hat{p}_{\mathcal{S}\cup i}(\boldsymbol{x}) - p(\boldsymbol{x}))^2 - \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup i}} p(\boldsymbol{x})^2 + \sum_{\boldsymbol{x}\in\mathcal{X}} p(\boldsymbol{x})^2 \qquad (7)
\end{aligned}$$

Noting that $p : \mathcal{X} \to \mathbb{R}_+$ is a distribution, we can explicitly find $\hat{p}_{\mathcal{S}\cup i}$, as shown in the main text, then

$$\sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup i}} (\hat{p}_{\mathcal{S}\cup i}(\boldsymbol{x}) - p(\boldsymbol{x}))^2 = \frac{(1 - \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup i}} p(\boldsymbol{x}))^2}{|\mathcal{X}_{\mathcal{S}\cup i}|}$$

Substituting it into equation (7), we have

$$\|\hat{p}_{\mathcal{S}\cup i} - p\|_2^2 = \frac{(1 - \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup i}} p(\boldsymbol{x}))^2}{|\mathcal{X}_{\mathcal{S}\cup i}|} - \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup i}} p(\boldsymbol{x})^2 + \sum_{\boldsymbol{x}\in\mathcal{X}} p(\boldsymbol{x})^2 \qquad (8)$$

We can see that the derivation of equation (8) also holds for $j \in \mathcal{S}'\backslash\mathcal{S}$, which means

$$\|\hat{p}_{\mathcal{S}\cup j} - p\|_2^2 = \frac{(1 - \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup j}} p(\boldsymbol{x}))^2}{|\mathcal{X}_{\mathcal{S}\cup j}|} - \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup j}} p(\boldsymbol{x})^2 + \sum_{\boldsymbol{x}\in\mathcal{X}} p(\boldsymbol{x})^2$$

In our discrete setting, we have $|\mathcal{X}_{\mathcal{S}\cup i}| = |\mathcal{X}_{\mathcal{S}\cup j}|$.

Therefore,

$$\begin{aligned}
&\|\hat{p}_{\mathcal{S}\cup i} - p\|_2^2 - \|\hat{p}_{\mathcal{S}\cup j} - p\|_2^2 \\
&= \frac{(1 - \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup i}} p(\boldsymbol{x}))^2 - (1 - \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup j}} p(\boldsymbol{x}))^2}{|\mathcal{X}_{\mathcal{S}\cup j}|} + \left(\sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup j}} p(\boldsymbol{x})^2 - \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup i}} p(\boldsymbol{x})^2\right) \\
&= \frac{\left(2 - \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup i}} p(\boldsymbol{x}) - \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup j}} p(\boldsymbol{x})\right)\left(\sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup j}} p(\boldsymbol{x}) - \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup i}} p(\boldsymbol{x})\right)}{|\mathcal{X}_{\mathcal{S}\cup j}|} \\
&\quad + \left(\sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup j}} p(\boldsymbol{x})^2 - \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup i}} p(\boldsymbol{x})^2\right)
\end{aligned}$$

Noting that p is a k-sparse distribution with support $\mathcal{S}'$, we can see that for $i \in [n]\backslash\mathcal{S}'$, $\{\mathcal{X}_{\mathcal{S}\cup i}\backslash\mathcal{X}_{\mathcal{S}}\} \cap \mathcal{X}_{\mathcal{S}'} = \emptyset$. Therefore, $p(\boldsymbol{x}) = 0$ where $\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}\cup i}\backslash\mathcal{X}_{\mathcal{S}}$. Hence,we can simplify the previous equation as

$$\begin{aligned}
&\|\hat{p}_{\mathcal{S}\cup i} - p\|_2^2 - \|\hat{p}_{\mathcal{S}\cup j} - p\|_2^2 \\
&= \frac{\left(2 - \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup i}} p(\boldsymbol{x}) - \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup j}} p(\boldsymbol{x})\right)\left(\sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup j}\backslash\mathcal{X}_{\mathcal{S}}} p(\boldsymbol{x})\right)}{|\mathcal{X}_{\mathcal{S}\cup j}|} + \left(\sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{S}\cup j}\backslash\mathcal{X}_{\mathcal{S}}} p(\boldsymbol{x})^2\right)
\end{aligned}$$

As we can see, if there exist $\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}\cup j} \backslash \mathcal{X}_{\mathcal{S}}$ for all $\mathcal{S} \subset \mathcal{S}', i \in [n] \backslash \mathcal{S}', j \in \mathcal{S}' \backslash \mathcal{S}$, such that $p(\boldsymbol{x}) > 0$, then $\|\hat{p}_{\mathcal{S}\cup i} - p\|_2^2 - \|\hat{p}_{\mathcal{S}\cup j} - p\|_2^2 > 0$ for all the $\mathcal{S}, i, j$. That is, conceptually, there are enough positions $\boldsymbol{x} \in \mathcal{X}_{\mathcal{S}'}$ where $p(\boldsymbol{x}) > 0$. And this condition leads us to the following wanted inequality.

$$\min_{\mathcal{S}:\mathcal{S}\subset\mathcal{S}'} \left(\min_{i\in[n]\backslash\mathcal{S}', j\in\mathcal{S}'\backslash\mathcal{S}} \|\hat{p}_{\mathcal{S}\cup i} - p\|_2 - \|\hat{p}_{\mathcal{S}\cup j} - p\|_2 \right) > 0. \tag{9}$$

Hence, under such conditions, *i.e.*, inequality (9) and inequality (6) hold, the greedy method is guaranteed to find exactly $\mathcal{S}'$.

□

G Proof of Theorem 4

Proof. Considering iteration t and $t+1$ as in algorithm 1. We drop parentheses for clarity. Applying RSS property we have:

$$\begin{aligned} F[p^{t+1}] - F[p^t] &\leq \left\langle \frac{\delta F}{\delta p^t}, p^{t+1} - p^t \right\rangle + \frac{\beta}{2}\|p^{t+1} - p^t\|_2^2 \\ &= \frac{1}{\mu}\langle p^t - q^{t+1}, p^{t+1} - p^t\rangle + \frac{\beta}{2}\|p^{t+1} - p^t\|_2^2 \end{aligned}$$

Setting step size $\mu = 1/\beta$, and then complete the square:

$$\begin{aligned} F[p^{t+1}] - F[p^t] &\leq \frac{\beta}{2}(\|p^{t+1} - q^{t+1}\|_2^2 - \|p^t - q^{t+1}\|_2^2) \\ &\leq \frac{\beta}{2}\left[(1+\phi)\|p^* - q^{t+1}\|_2^2 - \|p^t - q^{t+1}\|_2^2\right] \end{aligned}$$

where the inequality is due to approximate projection. Now by adding and subtracting p^t in $\|p^* - q^{t+1}\|_2^2$ on the right hand side, we have:

$$\begin{aligned} &\frac{\beta}{2}\left[(1+\phi)\left(\|p^* - p^t\|_2^2 + \|p^t - q^{t+1}\|_2^2 + 2\langle p^\star - p^t, p^t - q^{t+1}\rangle\right) - \|p^t - q^{t+1}\|_2^2\right] \\ &= \frac{\beta}{2}\left[(1+\phi)\|p^* - p^t\|_2^2 + \phi\|p^t - q^{t+1}\|_2^2 + 2(1+\phi)\langle p^\star - p^t, p^t - q^{t+1}\rangle\right] \\ &= \frac{\beta}{2}(1+\phi)\|p^* - p^t\|_2^2 + \frac{\phi}{2\beta}\|\frac{\delta F}{\delta p^t}\|_2^2 + (1+\phi)\langle p^\star - p^t, \frac{\delta F}{\delta p^t}\rangle \end{aligned}$$

Applying the Lipschitz condition 2, we have:

$$F[p^{t+1}] - F[p^t] \leq \frac{\beta}{2}(1+\phi)\|p^* - p^t\|_2^2 + \frac{\phi L^2}{2\beta} + (1+\phi)\langle p^\star - p^t, \frac{\delta F}{\delta p^t}\rangle \tag{10}$$

To bound the last inner product in (10), we need RSC property:

$$\langle p^\star - p^t, \frac{\delta F}{\delta p^t}\rangle \leq F[p^\star] - F[p^t] - \frac{\alpha}{2}\|p^t - p^\star\|_2^2$$

Apply it to relax the inner product term, we have (10):

$$\leq \frac{\beta - \alpha}{2}(1+\phi)\|p^\star - p^t\|_2^2 + \frac{\phi L^2}{2\beta} + (1+\phi)[F[p^\star] - F[p^t]] \tag{11}$$

Next we find the relation between $F[p^\star] - F[p^t]$ and $\|p^\star - p^t\|_2^2$ by RSC:

$$\begin{aligned} F[p^t] - F[p^\star] \geq \langle \frac{\delta F}{\delta p^\star}, p^t - p^\star\rangle + \frac{\alpha}{2}\|p^t - p^\star\|_2^2 \geq \frac{\alpha}{2}\|p^t - p^\star\|_2^2 - \|\frac{\delta F}{\delta p^\star}\|_2 \cdot \|p^t - p^\star\|_2 \\ \geq \frac{\alpha}{2}\|p^t - p^\star\|_2^2 - L\|p^t - p^\star\|_2 = \frac{\alpha}{2}\left[\|p^t - p^\star\|_2 - \frac{L}{\alpha}\right]^2 - \frac{L^2}{2\alpha} \end{aligned}$$

where the second inequality is by Cauchy–Schwarz inequality. When $\|p^t - p^\star\|_2 \leq \frac{L}{2\alpha}$, we have

$$F[p^t] - F[p^\star] \geq \frac{\alpha}{2}\|p^t - p^\star\|_2^2 - \frac{L^2}{2\alpha} \tag{12}$$

Apply (12) to (11) to convert $\|p^t - p^\star\|_2$ to $F[p^t] - F[p^\star]$, we have (11)

$$\leq (1+\phi)(2 - \frac{\beta}{\alpha})[F[p^\star] - F[p^t]] + \left(\frac{\phi}{2\beta} + (1+\phi)\frac{\beta - \alpha}{2\alpha^2}\right) L^2 \tag{13}$$

We denote the last term in (13) as c_1, and rearrange the equation, we have

$$F[p^{t+1}] - F[p^\star] \leq (1 - (1+\phi)(2 - \beta/\alpha)) \left[F[p^t] - F[p^\star]\right] + c_1$$

$$F[p^{t+1}] - F[p^\star] - c \leq (1 - (1+\phi)(2 - \beta/\alpha)) \left[F[p^t] - F[p^\star] - c\right] \tag{14}$$

where

$$c = \frac{c_1}{(1+\phi)(2-\beta/\alpha)} = \frac{\left(\phi/(2\beta) + (1+\phi)(\beta-\alpha)/(2\alpha^2)\right) L^2}{(1+\phi)(2-\beta/\alpha)}$$

From (14), we can see that if $0 < (1+\phi)(2-\beta/\alpha) < 1$, or $2 - 1/(1+\phi) < \beta/\alpha < 2$, IHT is guaranteed to converge to $F[p^\star] + c$ linearly. The smaller ϕ is and the closer is β to α, the smaller c is.

□

H Aditional Experimental details

Model Compression / Compressed sensing

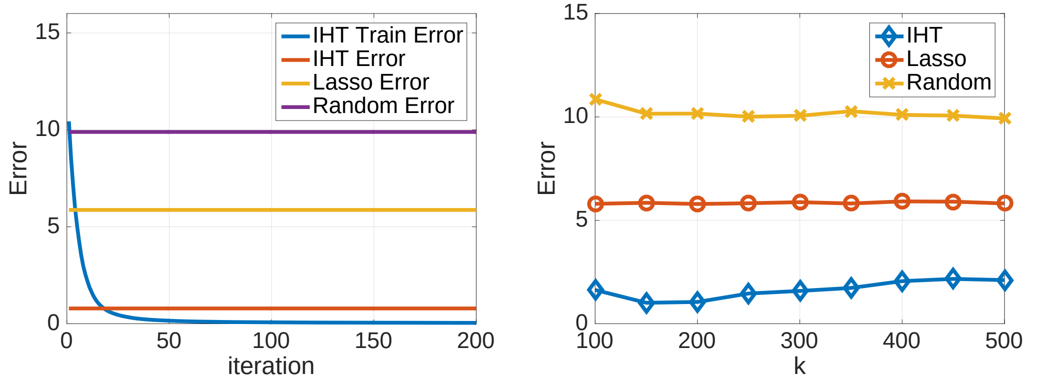
We apply our IHT to the task of expectation-preserving distribution compression, useful for efficiently storing large probability tables. Given a distribution $p(\cdot)$, our goal is to construct a sparse approximation $q(\cdot)$, such that $q(\cdot)$ approximately preserves expectations with respect to $p(\cdot)$. Interestingly, this model compression problem is equivalent to compressed sensing, but with the distributional constraints. We consider the vector sparsity in this experiment, *i.e.*, the distribution $\boldsymbol{q}$ is represented as a long vector in space $[0,1]^n$, as described in Appendix A. The problem setting we use in this experiment is to minimize $||A\boldsymbol{q} - A\boldsymbol{p}||_2^2$ subject to the vector distribution k-sparsity constraint of $\boldsymbol{q}$, *i.e.*, $||\boldsymbol{q}||_0 \leq k$ and $\sum_{i\in[n]} q_i = 1$. We first train the algorithms to minimize $||A\boldsymbol{q} - A\boldsymbol{p}||_2^2$ and then test their error on $||B\boldsymbol{q} - B\boldsymbol{p}||_2^2$, where A, B are randomly drawn from normal distribution $\mathcal{N}(0,1)$, and $\boldsymbol{p} \in [0,1]^n$ is a distribution generated from data.

We use real-world data: *Total Charges in 2012 Base Data 1, Hospital Discharge Data Public Use Data File*[1], which contains 740817 records. We set 10000 bins with bin size of 1000, to converge the data into a histogram, and hence the distribution $\boldsymbol{p} \in [0,1]^{10000}$. Note that $\boldsymbol{p}$ is already sparse. We set the dimension of A, B to 500×10000.

We compare IHT with two baselines, *i.e.*, Lasso and Random. Note that in vector-sparsity setting, the sparse l_2 distribution projection can be done optimally, since it becomes essentially a projection to a simplex, as we discussed in the main paper. The Lasso baseline is to minimize $||A\boldsymbol{q} - A\boldsymbol{p}||_2^2 + \gamma||\boldsymbol{q}||_1$ and then project its solution to the k-sparse distribution domain. The Random baseline is to randomly generate T k-sparse distribution, and simply choose the best, where T is the iteration number of IHT. Note that though the Greedy algorithm can work on this setting theoretically, it is too time costly to compare with the previously mentioned three algorithms.

As both training matrix A and testing matrix B are randomly generated, we use 10 different A for which we train the algorithms for 10 times, and after each training process we generate 20 test matrices B to test the error of each algorithms.

[1] https://www.dshs.state.tx.us/THCIC/Hospitals/Download.shtm



(a) IHT converges fast ($k = 200$)

(b) Error Comparison with different sparsity k

Figure 4: Real-data Experiments

Figure 4 (a) gives the convergence result when we set sparsity level $k = 200$. The IHT Train Error shows the training error of IHT at each iterations. IHT Error, Lasso Error and Random Error are testing errors of the three algorithms after training. We can see the promising results of IHT which outperforms other baselines. In Figure 4 (b), we test the three algorithms on different sparsity level $k = 100 \cdots 500$. IHT, Lasso and Random are testing errors of the three algorithms after training. Our results verify that IHT does the best regardless of sparsity level k.

Digits data: Dataset compression

We study representative prototype selection for the Digits data [31], which contains 7291 training and 2007 test examples of handwritten grayscale images. Prototypes are representative examples chosen from the data, in order to achieve dataset compression, while preserving certain desirable properties. In this experiment, our goal is to achieve compression to speed up nearest neighbor classification on unseen data as the quality measure of the selected prototypes. To this end, we embed the data using the RBF kernel $\exp(\gamma|\mathbf{x}_i - \mathbf{x}_i|^2)$, where the parameter γ is set using cross validation, and use the Maximum Mean Discrepancy (MMD) between the discrete data distribution in the embedded space, with the sparse data distribution representing the selected samples as our cost function for IHT. For two densities p and p, we can write $\text{MMD}^2 = E_{x,y\sim p}K(x,y) - 2E_{x\sim p,y\sim q}K(x,y) + E_{x\sim q,y\sim q}K(x,y)$, where $K(\cdot,\cdot)$ is the RBF kernel function. After the prototypes are selected, we evaluate the $0/1$ classification error with 1 Nearest Neighbor on the test data using only the selected prototypes.

We compare two forward selection greedy variants (Local Greedy and Global Greedy) proposed by [32] and the means algorithm (labeled as PS) proposed by [33], both state of the art. The results are presented in Figure 3(b). We see that IHT performs better than the baselines across different number of selected prototypes, especially when the number of prototypes is smaller.