Iterative Regularization with k-support Norm: An Important Complement to Sparse Recovery

William de Vazelhes[1], Bhaskar Mukhoty[1], Xiao-Tong Yuan[2], Bin Gu[1,3*]
[1]MBZUAI, Abu Dhabi, UAE
[2]Nanjing University, Suzhou, China
[3]Jilin University, Changchun, China
{wdevazelhes,bhaskar.mukhoty,xtyuan1980,jsgubin}@gmail.com

Abstract

Sparse recovery is ubiquitous in machine learning and signal processing. Due to the NP-hard nature of sparse recovery, existing methods are known to suffer either from restrictive (or even unknown) applicability conditions, or high computational cost. Recently, iterative regularization methods have emerged as a promising fast approach because they can achieve sparse recovery in one pass through early stopping, rather than the tedious grid-search used in the traditional methods. However, most of those iterative methods are based on the ℓ_1 norm which requires restrictive applicability conditions and could fail in many cases. Therefore, achieving sparse recovery with iterative regularization methods under a wider range of conditions has yet to be further explored. To address this issue, we propose a novel iterative regularization algorithm, IRKSN, based on the k-support norm regularizer rather than the ℓ_1 norm. We provide conditions for sparse recovery with IRKSN, and compare them with traditional conditions for recovery with ℓ_1 norm regularizers. Additionally, we give an early stopping bound on the model error of IRKSN with explicit constants, achieving the standard linear rate for sparse recovery. Finally, we illustrate the applicability of our algorithm on several experiments, including a support recovery experiment with a correlated design matrix.

Introduction

Sparse recovery is ubiquitous in machine learning and signal processing, with applications ranging from single pixel camera, to MRI, or radar[1]. In particular, with the ever-increasing amount of information, real-life datasets often contain much more features than samples: this is for instance the case in DNA microarray datasets (Golub et al. 1999), text data (Lang 1995), or image data such as fMRI (Belilovsky et al. 2015), where the number of features is generally much larger than the number of samples. In these high-dimensional settings, finding a linear model is under-specified, and therefore, one often needs to leverage additional assumptions about the true model, such as sparsity, to recover it. Usually, the problem is formulated as follows: we seek to recover a sparse vector $\boldsymbol{w}^* \in \mathbb{R}$ from its noisy linear measurements

$$\boldsymbol{y}^\delta = \boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\epsilon}$$

Here, $\boldsymbol{y}^\delta$ is a noisy measurement vector, i.e. a noisy version of the true target vector $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^*$, $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_d] \in \mathbb{R}^{n\times d}$ is a measurement matrix, also called design matrix, $\boldsymbol{\epsilon} \in \mathbb{R}^n$ is some bounded noise ($\|\boldsymbol{\epsilon}\|_2 \leq \delta$, with $\delta \in \mathbb{R}_+$), and $\boldsymbol{w}^*$ is the unknown k-sparse vector, i.e. containing only k nonzero components, that we wish to estimate with a vector $\hat{\boldsymbol{w}}$ obtained by running some sparse recovery algorithm on observations $\boldsymbol{y}^\delta$ and $\boldsymbol{X}$. Unfortunately, this problem is NP-hard in general, even in the noiseless setting (Natarajan 1995).

However, most of those iterative methods are based on the ℓ_1 norm which requires restrictive applicability conditions and could fail in many cases. We discuss such related works in more details in the next section. Therefore, achieving sparse recovery with iterative regularization methods under a wider range of conditions has yet to be further explored.

To address this issue, we propose a novel iterative regularization algorithm, IRKSN, based on the k-support norm regularizer rather than the ℓ_1 norm. That norm was first introduced in (Argyriou, Foygel, and Srebro 2012), as a way to improve upon the ElasticNet for sparse prediction. More precisely, we plug the k-support norm regularizer, for which there exist efficient proximal computations (Argyriou, Foygel, and Srebro 2012; McDonald, Pontil, and Stamos 2016b), into the primal-dual framework for iterative regularization described in (Matet et al. 2017).

We then provide some conditions for sparse recovery with IRKSN, and discuss on a simple example how they compare with traditional conditions for recovery with ℓ_1 norm regularizers.

More precisely, we elaborate on why such specific conditions include cases that are not included in some usual sufficient conditions for recovery with traditional methods based on the ℓ_1 norm (see Figure 1) (we describe such conditions for recovery with ℓ_1 norm in more details in Assumption 5). Since those types of conditions are still slightly opaque to interpret, we do as is common in the literature (such as in (Zou and Hastie 2005; Jia and Yu 2010)), namely, we discuss and compare those solutions with the help of an illustrative example. We also give an early stopping bound on the model error

*Corresponding author

[1]An introduction to this topic, as well as an extensive review of its applications can be found in (Foucart and Rauhut 2013) and (Wright and Ma 2022).

of IRKSN with explicit constants, achieving the standard linear rate for sparse recovery.

Finally, we illustrate the applicability of IRKSN on several experiments, including a support recovery experiment with a correlated design matrix, and show that it allows to identify the support more accurately than its competitors.

Contributions. We summarize the main contributions of our paper as follows:

1. We introduce a new algorithm, IRKSN, which allows recovery of the true sparse vector under conditions for which some sufficient conditions for recovery with ℓ_1 norm do not hold. We discuss the difference between those conditions on a detailed example.
2. We give an early stopping bound on the model error of IRKSN with explicit constants, achieving the standard linear rate for sparse recovery.
3. We illustrate the applicability of our algorithm on several experiments, including a support recovery experiment with a correlated design matrix, and show that it allows support recovery with a higher F1 score than its competitors.

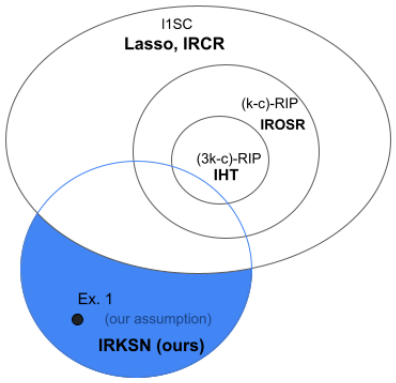


Figure 1: Conditions for recovery in various settings: l1SC corresponds to the condition $\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \operatorname{sgn}(\boldsymbol{w}_S^*)\rangle| < 1$. "ours" denotes the condition $\max_{i \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_i, \boldsymbol{w}_S^*\rangle| < \min_{j \in S} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_j, \boldsymbol{w}_S^*\rangle|$. c denotes some constant in $[0, 1]$. Here $3k$-RIP is shown for indicative purposes, corresponding to the condition for IHT as described in (Blumensath and Davies 2009). As we can see, for some cases (in blue), only IRKSN (our algorithm) can provably ensure sparse recovery.

Preliminaries

Notations. We first recall a few definitions and notations used in the rest of the paper. We denote all vectors and matrices variables in bold font. For $S \subseteq [d]$, $\bar{S}$ denotes $[d] \setminus S$. For any matrix $\boldsymbol{M} \in \mathbb{R}^{n \times d}$, $\boldsymbol{m}_i$ denotes its i-th column for $i \in \mathbb{N}$, $\boldsymbol{M}^\top$ its transpose, $\boldsymbol{M}^\dagger$ its Moore-Penrose pseudo-inverse (Golub and Van Loan 2013), $\|\boldsymbol{M}\|$ its nuclear norm, and $\boldsymbol{M}_S$ its column-restriction to a support $S \subseteq [d]$, i.e. the $n \times |S|$ matrix composed of the $|S|$ columns of $\boldsymbol{M}$ of indices in S. For a vector $\boldsymbol{w} \in \mathbb{R}^d$, $\operatorname{supp}(\boldsymbol{w})$ denotes its support $\boldsymbol{w}$, that is, the coordinates of the non-zero components of $\boldsymbol{w}$, w_i denotes its i-th component, $|w|_i^\downarrow$ denotes its i-th top absolute value, and $\|\boldsymbol{w}\|$ denotes its ℓ_2 norm.

More generally $\|\boldsymbol{w}\|_p$ denotes its ℓ_p norm for $p \in [1, +\infty)$, and $\|\boldsymbol{w}\|_0$ denotes its number of non-zero components. $\boldsymbol{w}_S \in \mathbb{R}^k$ denotes its restriction to a support S of size k, that is, the sub-vector of size k formed by extracting only the components w_i with $i \in S$. $\operatorname{sgn}(\boldsymbol{w})$ denotes the vector of its signs (with the additional convention that if $\boldsymbol{w}_i = 0$, $\operatorname{sgn}(\boldsymbol{w})_i = 0$).

Related works. Due to the NP-hard nature of sparse recovery, existing methods are known to suffer either from restrictive (or even unknown) applicability conditions, or high computational cost. Amongst those methods, a first group of methods can achieve an exact sparsity k of the estimate $\hat{\boldsymbol{w}}$: Iterative Hard Thresholding (Blumensath and Davies 2009) returns an estimate $\hat{\boldsymbol{w}}$ which recovers $\boldsymbol{w}^*$ up to an error $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\| \leq O(\delta)$, if the design matrix $\boldsymbol{X}$ satisfies some Restricted Isometry Property (RIP) (Blumensath and Davies 2009). However, as mentioned in (Jain, Tewari, and Kar 2014), this condition is very restrictive, and does not hold in most high-dimensional problems. Greedy methods, such as Orthogonal Matching Pursuit (OMP) (Tropp and Gilbert 2007), also can return an exactly k-sparse vector, and bounds on the recovery of a (generalized version of) OMP, of the type $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\| \leq O(\delta)$, can be found for instance in (Wang et al. 2015), under some RIP condition.

A second set of methods for sparse recovery solve the following penalized problem:

$$(P) : \min_{\boldsymbol{w}} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}^\delta\|^2 + \lambda R(\boldsymbol{w})$$

Where R is a regularizer, such as the ℓ_1 norm as is done in the Lasso method (Tibshirani 1996), and λ is a penalty parameter that needs to be tuned. For a given λ, (P) is usually solved through a convex optimization algorithm, and returns a solution $\hat{\boldsymbol{w}}$ of (P), as an estimate of $\boldsymbol{w}^*$. Amongst those, one of the most important algorithms for sparse recovery, the Lasso (Tibshirani 1996), has been proven in (Grasmair, Scherzer, and Haltmeier 2011) to give a bound $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\| \leq O(\delta)$ under the so-called *source conditions* (described in Condition 4.3 from (Grasmair, Scherzer, and Haltmeier 2011)) which are implied by the following more intuitive conditions: $\boldsymbol{X}_S$ is injective, and $\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \operatorname{sgn}(\boldsymbol{w}_S^*)\rangle| < 1$ (we detail this implication in Assumption 5). Following the Lasso, the ElasticNet was later developed to solve the problem of a design matrix with possibly high correlations. However, although some conditions for *statistical consistency* exist for the ElasticNet (Jia and Yu 2010), to the best of our knowledge, there is no model error bound (and conditions thereof) for recovery with ElasticNet. Finally, the k-support norm regularization has also been used successfully as a penalty (Argyriou, Foygel, and Srebro 2012), with even better empirical results than the ElasticNet, but no explicit error bounds on model error (and the conditions thereof) currently exists: indeed, their work was mostly focused on *sparse prediction* and not *sparse recovery*. Efficient solvers have later been derived for the Lasso using for instance coordinate descent and its

METHOD	CONDITION ON $\boldsymbol{X}$	BOUND ON $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\|$	COMPLEXITY
IHT (BLUMENSATH AND DAVIES 2009)	RIP	$O(\delta)$	$O(T)$
LASSO (TIBSHIRANI 1996)	$\max_{\ell \in \bar{S}} \lvert\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \operatorname{sgn}(\boldsymbol{w}_S^*)\rangle\rvert < 1$ (2)	$O(\delta)$	$O(\Lambda T)$
ELASTICNET (ZOU AND HASTIE 2005)	-	-	$O(\Lambda T)$
KSN PEN. (ARGYRIOU, FOYGEL, AND SREBRO 2012)	-	-	$O(\Lambda T)$
OMP (TROPP AND GILBERT 2007)	RIP	$O(\delta)$	$O(k)$
SRDI (OSHER ET AL. 2016)	$\begin{cases} \exists \gamma \in (0,1]: \ \boldsymbol{X}_S^\top \boldsymbol{X}_S \geq n\gamma I_{d,d} \\ \exists \eta \in (0,1): \ \|\boldsymbol{X}_{\bar{S}} \boldsymbol{X}_S^\dagger\|_\infty \leq 1-\eta \end{cases}$	$O(\sigma\sqrt{\frac{k \log d}{n}})$ (1)	$O(T)$
IROSR (VASKEVICIUS, KANADE, AND REBESCHINI 2019)	RIP	$O(\sigma\sqrt{\frac{k \log d}{n}})$ (1)	$O(T)$
IRCR (MOLINARI ET AL. 2021)	$\max_{\ell \in \bar{S}} \lvert\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \operatorname{sgn}(\boldsymbol{w}_S^*)\rangle\rvert < 1$ (2)	$O(\delta)$	$O(T)$
IRKSN (OURS)	$\max_{\ell \in \bar{S}} \lvert\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^*\rangle\rvert < \min_{j \in S} \lvert\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_j, \boldsymbol{w}_S^*\rangle\rvert$	$O(\delta)$	$O(T)$

Table 1: Comparison of the existing algorithms for sparse recovery in the literature, including conditions on $\boldsymbol{X}$ and $\boldsymbol{w}^*$ sufficient for recovery. T is the number of iterations each algorithm is ran for, and Λ is the number of values of λ that need to be tried out (for penalized methods). (1) assuming $\epsilon \underset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$. (2): Additionally, $\boldsymbol{X}_S$ should be injective.

variants (Fang et al. 2020; Bertrand and Massias 2021). However, even with efficient solvers, these penalized methods need to tune the parameter λ, which is very costly.

Recently, iterative regularization methods have emerged as a promising fast approach because they can achieve sparse recovery in one pass through early stopping, rather than the tedious grid-search used in traditional methods. They solve the following problem

$$(I): \quad \min_{\boldsymbol{w}} R(\boldsymbol{w}) \quad \text{s.t.} \quad \boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}^\delta$$

An iterative algorithm is used to solve it, and returns some $\hat{\boldsymbol{w}}$ to estimate $\boldsymbol{w}^*$. Importantly, $\hat{\boldsymbol{w}}$ is obtained by stopping the algorithm before convergence, also called *early stopping*. One of the first amongst these methods, SRDI (Osher et al. 2016), achieves a rate of $\|\hat{\boldsymbol{w}} - \boldsymbol{w}\| \leq O(\sigma\sqrt{\frac{k \log d}{n}})$ with high probability, assuming $\epsilon \underset{i.i.d.}{\sim} \mathcal{N}(0, \sigma)$, and two conditions: (1) $\exists \gamma \in (0,1]: \ \boldsymbol{X}_S^\top \boldsymbol{X}_S \geq n\gamma I_{d,d}$ (Restricted Strong Convexity) and (2) $\exists \eta \in (0,1): \ \|\boldsymbol{X}_{\bar{S}} \boldsymbol{X}_S^\dagger\|_\infty \leq 1-\eta$. IROSR (Vaskevicius, Kanade, and Rebeschini 2019) uses an iterative regularization scheme that is based on a reparameterization of the problem (I). They prove a high probability model consistency bound of $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^*\| \leq O(\sigma\sqrt{\frac{k \log d}{n}})$, assuming the $((k+1, c)$-RIP for some constant $c(k, \boldsymbol{w}^*, \boldsymbol{X}, \epsilon)$. Similar to their work is (Zhao, Yang, and He 2022): under similar conditions, they also obtain a similar rate. Finally, (Molinari et al. 2021) provide bounds of the form $\|\hat{\boldsymbol{w}} - \boldsymbol{w}\| \leq O(\delta)$, under the same *source conditions* as in (Grasmair, Scherzer, and Haltmeier 2011).

However, most of those iterative methods are based on the ℓ_1 norm which requires restrictive applicability conditions and could fail in many cases. Indeed, in those cases, the conditions for recovery with the methods described above (e.g. RIP, or the sufficient conditions for recovery with Lasso that we discussed above) do not hold anymore. For instance, in gene array data (Zou and Hastie 2005), it is known that many columns of the design matrix are correlated, and that RIP does not hold. It is therefore crucial to come up with algorithms for which recovery is provably possible under different conditions, which we tackle in this paper.

k-support Norm Regularization. We now introduce the k-support norm, which is the main component of our algorithm, as well as its proximal operator. The k-support norm was first introduced in (Argyriou, Foygel, and Srebro 2012), as the tightest convex relaxation of the intersection of the ℓ_2 ball and the ℓ_0 ball. It was later generalized to the matrix case (McDonald, Pontil, and Stamos 2016a,b), as well as successfully applied to several problems, including for instance fMRI (Gkirtzou et al. 2013; Belilovsky et al. 2015). We give below its formal definition, with the following variational formula from (Argyriou, Foygel, and Srebro 2012):

Definition 1 ((Argyriou, Foygel, and Srebro 2012; McDonald, Pontil, and Stamos 2014))**.** Let $k \in \{1, ..., d\}$. The k-support norm $\|\cdot\|_k^{sp}$ is defined, for every $\boldsymbol{w} \in \mathbb{R}^d$, as:

$$\|\boldsymbol{w}\|_k^{sp} = \min \left\{ \sum_{I \in \mathcal{G}_k} \|\boldsymbol{v}_I\|_2 : \boldsymbol{v}_I \in \mathbb{R}^d, \operatorname{supp}(\boldsymbol{v}_I) \subseteq I, \sum_{I \in \mathcal{G}_k} \boldsymbol{v}_I = \boldsymbol{w} \right\}$$

where $\mathcal{G}_k$ denotes the set of all subsets of $\{1, ..., d\}$ of cardinality at most k.

In other words, the k-support norm is equal to the smallest sum of the norms of some k-sparse *atoms* (the $\boldsymbol{y}_I$ above) that constitute $\boldsymbol{w}$: as studied in (Chatterjee, Chen, and Banerjee 2014), the k-support norm is indeed a so-called *atomic norm*. One can also see from this definition that the k-support norm interpolates between the ℓ_1 norm (which it is equal to if $k = 1$) and the ℓ_2 norm (which it is equal to if $k = d$). As

discussed in (Argyriou, Foygel, and Srebro 2012), another interpretation of the k-support norm is that it is equivalent to the Group-Lasso penalty with overlaps (Jacob, Obozinski, and Vert 2009), when the set of overlapping groups is all possible subsets of $\{1, ..., d\}$ of cardinality at most k. Finally, we introduce the proximal operator (Parikh, Boyd et al. 2014) below, that will be used in our algorithm:

Definition 2 (Proximal operator, (Parikh, Boyd et al. 2014))**.** The proximal operator for a function $h : \mathbb{R}^d \to \mathbb{R}$ is defined as:

$$\text{prox}_h(\boldsymbol{z}) = \arg\min_{\boldsymbol{w}} h(\boldsymbol{w}) + \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{z}\|_2^2$$

A closed form for the proximal operator of the squared k-support norm was first given in (Argyriou, Foygel, and Srebro 2012), and more efficient computations have been found e.g. in (McDonald, Pontil, and Stamos 2016b), which we will use in IRKSN, as described in Appendix E.

The Algorithm

In this section, we describe the IRKSN (Iterative Regularization with k-Support Norm) algorithm. It is based on the general accelerated algorithm from (Matet et al. 2017), in which we plug a regularization function based on the k-support norm. More precisely, (Matet et al. 2017) describe a general regularization algorithm for model recovery based on a primal-dual method, and an early stopping rule. As they do, we will solve the following problem approximately (i.e. with early stopping):

$$(I_{ks}): \quad \min_{\boldsymbol{w}} R(\boldsymbol{w}) \quad \text{s.t.} \quad \boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}^\delta$$

with a specific regularizer that we introduce: $R(\boldsymbol{w}) = F(\boldsymbol{w}) + \frac{\alpha}{2}\|\boldsymbol{w}\|_2{}^2$ with $F(\boldsymbol{w}) = \frac{1-\alpha}{2}(\|\boldsymbol{w}\|_k^{sp})^2$, for some constant $1 > \alpha > 0$ which will be described later. The algorithm that we will use to solve approximately (I_{ks}) is the Accelerated Dual Gradient Descent (ADGD) described in (Matet et al. 2017), which is an accelerated version of a primal-dual method that is known in the literature under many names, and that comprises the following steps, with γ being some learning rate, and $\hat{\boldsymbol{v}}_t$ being a dual variable:

`# primal projection step`
$\hat{\boldsymbol{w}}_t \leftarrow \text{prox}_{\alpha^{-1}F}(-\alpha^{-1}\boldsymbol{X}^\top \hat{\boldsymbol{v}}_t)$
`# dual update step`
$\hat{\boldsymbol{v}}_{t+1} \leftarrow \hat{\boldsymbol{v}}_t + \gamma(\boldsymbol{X}\hat{\boldsymbol{w}}_t - \boldsymbol{y}^\delta)$

The method above is most commonly known in the signal processing and image denoising literature as Linearized Bregman Iterations, or Inverse Scale Space Methods (Cai, Osher, and Shen 2009; Osher et al. 2016). In the optimization literature, it is mostly known as (Lazy) Mirror Descent (Bubeck et al. 2015), also called Dual Averaging (Nesterov 2009; Xiao 2009). The main idea in (Matet et al. 2017) is to early stop the algorithm at some iteration T, before convergence. We present the full accelerated version, IRKSN, in Algorithm 1.

Main Results

In this section, we introduce the main result of our paper, which gives specific conditions for robust recovery of $\boldsymbol{w}^*$, and early stopping bounds on $\|\hat{\boldsymbol{w}}_t - \boldsymbol{w}^*\|$ for IRKSN.

Assumptions

We will present several sufficient conditions for recovery with the k-support norm, which are similar to the sufficient conditions needed for ℓ_1-based recovery that we describe in Assumption 5 (we will then elaborate on the differences between such conditions). The first assumption below is a variant of the usual feasibility assumption of the noiseless problem (Foucart and Rauhut 2013): it simply states that $\boldsymbol{w}^*$, the true model that we wish to recover, is a feasible solution of the noiseless problem, and that it is k-sparse. Additionally, if several feasible solutions of same support than $\boldsymbol{w}^*$ exist, $\boldsymbol{w}^*$ should be the smallest norm one (we will elaborate on such condition in this section). Recall from the Introduction that $\boldsymbol{y}$ is the true target vector, i.e. uncorrupted by noise.

Assumption 3. $\boldsymbol{w}^*$ is k-sparse of support $S \subset [d]$, and is a solution of the system $(L) : \boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}$. In addition, $\boldsymbol{w}^*$ is the smallest ℓ_2 norm solution of (L) on its support, that is, $\boldsymbol{w}^*$ is such that:

$$\boldsymbol{w}_S^* = \arg \min_{\boldsymbol{z} \in \mathbb{R}^k : \boldsymbol{X}_S \boldsymbol{z} = \boldsymbol{y}} \|\boldsymbol{z}\|_2$$

We now provide our main assumption, which is intrinsically linked to the structure of the k-support norm, and which is, up to our knowledge, the first condition of such kind in the sparse recovery literature.

Assumption 4. $\boldsymbol{w}^*$ verifies:

$$\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^* \rangle| < \min_{j \in S} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_j, \boldsymbol{w}_S^* \rangle|$$

Up to our knowledge, we are the first to provide such assumptions for recovery with a k-support norm based algorithm: although (Chatterjee, Chen, and Banerjee 2014) proposed a k-support norm based algorithm and corresponding conditions for recovery, those conditions only apply in the case of a design matrix $\boldsymbol{X}$ with values which are i.i.d. samples from a Gaussian distribution.

Discussion on the Assumptions

In this section, we attempt to interpret the assumptions above in simple terms, and to compare them to some similar sufficient conditions for recovery with ℓ_1 norm. More precisely, the condition below implies Condition 4.3 from (Grasmair, Scherzer, and Haltmeier 2011), which latter is shown in (Grasmair, Scherzer, and Haltmeier 2011) to be a necessary and sufficient condition for achieving a linear rate of recovery with ℓ_1 norm Tikhonov regularization. We prove such implication in Appendix B.

Algorithm 1: IRKSN

Input: $\hat{\boldsymbol{v}}_0 = \hat{\boldsymbol{z}}_{-1} = \hat{\boldsymbol{z}}_0 \in \mathbb{R}^d, \gamma = \alpha\|\boldsymbol{X}\|^{-2}, \theta_0 = 1$
for $t = 0$ **to** T **do**
$\hat{\boldsymbol{w}}_t \leftarrow \text{prox}_{\alpha^{-1}F}\left(-\alpha^{-1}\boldsymbol{X}^T\hat{\boldsymbol{z}}_t\right)$
$\hat{\boldsymbol{r}}_t \leftarrow \text{prox}_{\alpha^{-1}F}\left(-\alpha^{-1}\boldsymbol{X}^T\hat{\boldsymbol{v}}_t\right)$
$\hat{\boldsymbol{z}}_t \leftarrow \hat{\boldsymbol{v}}_t + \gamma\left(\boldsymbol{X}\hat{\boldsymbol{r}}_t - \boldsymbol{y}^\delta\right)$
$\theta_{t+1} \leftarrow \left(1 + \sqrt{1 + 4\theta_t^2}\right)/2$
$\hat{\boldsymbol{v}}_{t+1} = \hat{\boldsymbol{z}}_t + \frac{\theta_t - 1}{\theta_{t+1}}\left(\hat{\boldsymbol{z}}_t - \hat{\boldsymbol{z}}_{t-1}\right)$
end for

Assumption 5 (Recovery with ℓ_1 norm.)**.** Let $\boldsymbol{w}^*$ be supported on a support $S \subset [d]$. $\boldsymbol{w}^*$ is such that:

(i) $\boldsymbol{X}\boldsymbol{w}^* = \boldsymbol{y}$
(ii) $\boldsymbol{X}_S$ is injective
(iii) $\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \operatorname{sgn}(\boldsymbol{w}_S^*)\rangle| < 1$

Below, we now compare this assumption to ours.

The min ℓ_2 norm solution. In our Assumption 3, the minimum ℓ_2 norm condition is actually not restrictive, compared to Assumption 5: indeed, in Assumption 5 $\boldsymbol{X}_S$ needs to be injective, which implies that there needs to be only one solution $\boldsymbol{w}_S^*$ on S such that $\boldsymbol{X}_S \boldsymbol{w}_S^* = \boldsymbol{y}$: we can also work in such situations, but we also include the additional cases where there are several solutions on S (we just require that $\boldsymbol{w}^*$ is the minimum norm one) : $\boldsymbol{X}_S$ does not need to be injective in our case. Importantly we can deal with cases with $n < k$, when Lasso (and ℓ_1 iterative regularization methods) cannot (that is, we can obtain recovery in a regime where the number of samples n is even lower than the sparsity of the signal k). Note that for the Lasso, the condition $n \geq k$ is even *necessary*: indeed, when $n < k$, the Lasso is known to saturate (Zou and Hastie 2005) and recovery is impossible: interestingly, there is no such constraint when using a k-support norm regularizer (similarly to recovery with ElasticNet).

Dependence on the sign. As we can observe, Assumption 5 is verified or not based on $\operatorname{sgn}(\boldsymbol{w}_S^*)$. This implies that irrespective of the actual values of $\boldsymbol{w}^*$, recovery will be possible or not only based on $\operatorname{sgn}(\boldsymbol{w}_S^*)$. On the contrary, our Assumption 4 depends on $\boldsymbol{w}^*$ itself.

Case where $\boldsymbol{X}_S$ is injective. In the case where $\boldsymbol{X}_S$ is injective (as will happen in most cases in practice when $n > k$, i.e. unless there is some spurious exact linear dependence between columns), it is even easier to compare Assumptions 4 and 5. Indeed, since in that case we have that $\boldsymbol{X}_S$ is full column rank, we then have : $\boldsymbol{X}_S^\dagger \boldsymbol{X}_S = \boldsymbol{I}_{k\times k}$. Therefore, Assumption 4 can be rewritten into: $\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \boldsymbol{w}_S^*\rangle| < \min_{j\in S} |w_i^*|$, which is equivalent to:

$$\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \frac{\boldsymbol{w}_S^*}{\min_{j\in S} |w_i^*|}\rangle| < 1$$

Therefore, we can notice that if $\boldsymbol{w}_S^* = \gamma \operatorname{sgn}(\boldsymbol{w}_S^*)$ for some $\gamma > 0$ (that is, each component of $\boldsymbol{w}_S^*$ have the same absolute value), both Assumptions 4 and 5 become equivalent (because then: $\frac{\boldsymbol{w}_S^*}{\min_{j\in S} |w_i^*|} = \operatorname{sgn}(\boldsymbol{w}_S^*)$). However, the two conditions 4 and 5 may differ depending on the *relative magnitudes* of the entries in $\boldsymbol{w}_S^*$. In particular, it may happen that our Assumption 4 is verified even if the Assumption 5 is not verified. We analyze such an example in Example 1.

Early Stopping Bound

We are now ready to state our main result:

Theorem 6 (Early Stopping Bound)**.** *Let $\delta \in]0, 1]$ and let $(\hat{\boldsymbol{w}}_t)_{t\in\mathbb{N}}$ be the sequence generated by IRKSN. Assuming the design matrix $\boldsymbol{X}$ and the true sparse vector $\boldsymbol{w}^*$ satisfy Assumptions 3 and 4, and with $\alpha < \frac{\eta}{\|\boldsymbol{w}\|_\infty}$ with $\eta := \min_{j\in S} |\langle (\boldsymbol{X}_S \boldsymbol{X}_S^\top)^\dagger \boldsymbol{y}, \boldsymbol{x}_j\rangle| - \max_{\ell\in\bar{S}} |\langle (\boldsymbol{X}_S \boldsymbol{X}_S^\top)^\dagger \boldsymbol{y}, \boldsymbol{x}_\ell\rangle|$, we have for $t \geq 2$:*

$$\|\hat{\boldsymbol{w}}_t - \boldsymbol{w}^*\|_2 \leq at\delta + bt^{-1}$$

$$\textit{with} \quad a = 4\|\boldsymbol{X}\|^{-1} \quad \textit{and} \quad b = \frac{2\|\boldsymbol{X}\|\|(\boldsymbol{X}_S^\top)^\dagger \boldsymbol{w}_S^*\|}{\alpha}$$

In particular (if $\delta > 0$), with $t_\delta = \lceil c\delta^{-1/2} \rceil$, for some $c > 0$:

$$\|\hat{\boldsymbol{w}}_t - \boldsymbol{w}^*\|_2 \leq (a(c+1) + bc^{-1})\delta^{1/2}$$

Proof. Proof in Appendix C. □

Discussion. We can notice in Theorem 6 above that b is large when α is small: therefore, if the inequality in 4 is very tight, as a consequence, α will need to be taken small, and b will become large. Therefore, we can say that the larger the margin by which Assumption 4 is fulfilled is, the better the retrieval of the true vector $\boldsymbol{w}^*$ is (because the larger we can choose α).

Illustrating Example

In this section, we describe a simple example that illustrates the cases where ℓ_1 norm-based regularization fails, and where IRKSN will successfully recover the true vector.

Example 1. We consider a model that consists of three "generating" variables $X^{(0)}, X^{(1)}$ and $X^{(2)}$, that are random i.i.d. variables from standard Gaussian (we denote $X^{(0)} \sim \mathcal{N}(0, 1)$ and $X^{(1)} \sim \mathcal{N}(0, 1)$ and $X^{(2)} \sim \mathcal{N}(0, 1)$). Two other variables $X^{(3)}$ and $X^{(4)}$, are actually correlated with the previous random variables: they are obtained noiselessly, and linearly from those, with some vectors $\boldsymbol{w}^{(3)}$ and $\boldsymbol{w}^{(4)}$ that will be defined below:

$$X^{(3)} = w_0^{(3)} X^{(0)} + w_1^{(3)} X^{(1)} + w_2^{(3)} X^{(2)}$$

and

$$X^{(4)} = w_0^{(4)} X^{(0)} + w_1^{(4)} X^{(1)} + w_2^{(4)} X^{(2)}$$

In addition, similarly, the actual observations Y are formed noiselessly and linearly from $(X^{(0)}, X^{(1)}, X^{(2)})$, for some vector $\boldsymbol{w}^{(y)}$:

$$Y = w_0^{(y)} X^{(0)} + w_1^{(y)} X^{(1)} + w_2^{(y)} X^{(2)}$$

A graphical visualization of this construction can be seen on Figure 2. More precisely, we define the vectors $\boldsymbol{w}^{(3)}, \boldsymbol{w}^{(4)}$ and $\boldsymbol{w}^{(y)}$ are defined as follows:

$$\boldsymbol{w}^{(3)} = \begin{bmatrix} 9/11 \\ 6/11 \\ 2/11 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{w}^{(4)} = \begin{bmatrix} 1/3 \\ 14/15 \\ 2/15 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{w}^{(y)} = \begin{bmatrix} 1 \\ 1 \\ -4 \\ 0 \\ 0 \end{bmatrix}.$$

We will generate such a dataset with $n = 4$: so the dataset will be composed of 4 samples of $X^{(0)}, X^{(1)}, X^{(2)}, X^{(3)}, X^{(4)}$, which form the matrix $\boldsymbol{X} \in \mathbb{R}^{4,5}$, with $\boldsymbol{X} = [\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4]$ and 4 samples of Y, which form the vector $\boldsymbol{y} \in \mathbb{R}^4$. In our case, we have $S = \operatorname{supp}(\boldsymbol{w}^{(y)}) = \{0, 1, 2\}$, and therefore we just ensure that $\boldsymbol{X}_S = [\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2]$ is full column rank (which should be the case with overwhelming probability since those three first vectors are sampled from a Gaussian, and since we have $n = 4 > k = 3$). Our goal is to reconstruct the true linear model of Y, which is $\boldsymbol{w}^{(y)}$ from the observation of $\boldsymbol{X}$ and $\boldsymbol{y}$. We can eas-

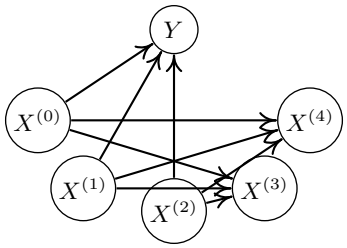


Figure 2: $X^{(3)}$, $X^{(4)}$ are correlated with $X^{(0)}, X^{(1)}, X^{(2)}$

ily check mathematically (using the closed form from the first column of Table 1), that this example only verifies our condition (Assumption 4), but that it does not verify Assumption 5 (i.e. it is in the blue area from Figure 1). Indeed, in that case, $\boldsymbol{X}_S$ is full column rank, which implies $(\boldsymbol{X}_S)^\dagger \boldsymbol{x}_3 = \boldsymbol{w}^{(3)}$ and $(\boldsymbol{X}_S)^\dagger \boldsymbol{x}_4 = \boldsymbol{w}^{(4)}$ (Golub and Van Loan 2013). We then have:

$$|\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_3, \operatorname{sgn}(\boldsymbol{w}^{(y)})\rangle| = |\langle \boldsymbol{w}^{(3)}, \operatorname{sgn}(\boldsymbol{w}^{(y)})\rangle| = 13/11 > 1$$

$$|\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_4, \operatorname{sgn}(\boldsymbol{w}^{(y)})\rangle| = |\langle \boldsymbol{w}^{(4)}, \operatorname{sgn}(\boldsymbol{w}^{(y)})\rangle| = 17/15 > 1$$

Therefore: $\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \operatorname{sgn}(\boldsymbol{w}_S^*)\rangle| = \frac{13}{11} > 1$ Which means that Assumption 5 is not verified. However, on the other hand, we have:

$$|\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_3, \frac{\boldsymbol{w}^{(y)}}{\min_{j\in S} |w_i^{(y)}|}\rangle| = |\langle \boldsymbol{w}^{(3)}, \frac{\boldsymbol{w}^{(y)}}{\min_{j\in S} |w_i^{(y)}|}\rangle| = \frac{7}{11}$$

$$|\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_4, \frac{\boldsymbol{w}^{(y)}}{\min_{j\in S} |w_i^{(y)}|}\rangle| = |\langle \boldsymbol{w}^{(4)}, \frac{\boldsymbol{w}^{(y)}}{\min_{j\in S} |w_i^{(y)}|}\rangle| = \frac{11}{15}$$

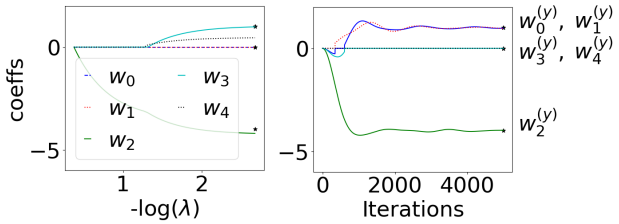
Therefore: $\max_{\ell \in \bar{S}} |\langle \boldsymbol{X}_S^\dagger \boldsymbol{x}_\ell, \frac{\boldsymbol{w}^{(y)}}{\min_{j\in S} |w_i^{(y)}|}\rangle| = \frac{11}{15} < 1.$ Therefore, from the Section *Discussion on the Assumptions*, paragraph *Case where $\boldsymbol{X}_S$ is injective*, we see that our Assumption 4 is verified here.

Comparison of the IRKSN path with Lasso. In Figure 3 below, we compare the Lasso path (that is, the solutions found by Lasso for all values of the penalization λ), with the IRKSN path (that is, the solutions found by IRKSN at every timestep). For indicative purposes, we also provide the path of the ElasticNet on the same problem in Appendix G.

Experiments

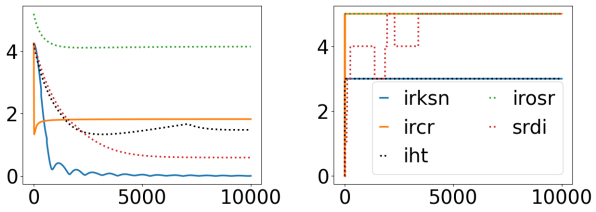
As we can see, the Lasso is unable to retrieve the true sparse vector, for any λ. However IRKSN can successfully retrieve it, which confirms the theory above.

In addition, this path from Figure 3 above illustrates well the optimization dynamics of IRKSN: first, the true support of $\boldsymbol{w}^{(y)}$ is not identified in the first iterations. But after a few iterations, we observe what we could call a phenomenon of *exchange of variable*: $w_0^{(y)}$ is exchanged with $w_1^{(y)}$, and later, $w_3^{(y)}$ is exchanged with $w_0^{(y)}$ (by *exchange*, we mean that at a timestep t, $w_0^{(y)}(t) \neq 0$ but $w_1^{(y)}(t) = 0$, but at



(a) Lasso path (b) IRKSN path

Figure 3: Comparison of the path of IRKSN with Lasso. $w_i^{(y)}$ is the i-th component of $\boldsymbol{w}^{(y)}$, and λ is the penalty of the Lasso. We recall $w_0^{(y)} = w_1^{(y)} = 1, w_2^{(y)} = -4, w_3^{(y)} = w_4^{(y)} = 0$: only IRKSN recovers the true $\boldsymbol{w}^{(y)}$.



(a) Model error $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^{(y)}\|$ (b) Model sparsity $\|\hat{\boldsymbol{w}}\|_0$

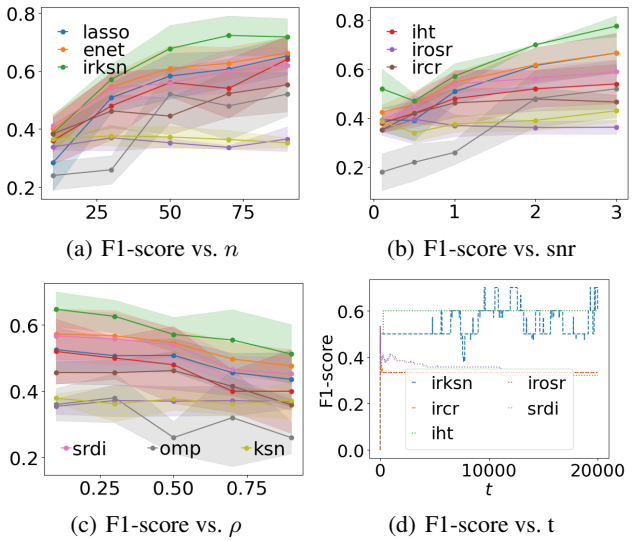
Figure 4: Error and sparsity vs. number of iterations. Only IRKSN can recover the true $\boldsymbol{w}^{(y)}$ in this example.

timestep $t+1$: $w_0^{(y)}(t+1) = 0$ and $w_1^{(y)}(t+1) \approx w_0^{(y)}(t)$). This can be explained by the fact that when α is small, the proximal operator of the k-support norm approaches the hard-thresholding operator from (Blumensath and Davies 2009): hence at a particular timestep the ordering (in absolute magnitude) of the components of $\boldsymbol{X}^\top \hat{\boldsymbol{z}}_t$ suddenly changes (with the components where the change occurs having about the same magnitude at the time of change, if the learning rate is small), which results into such an observed change in primal space. Additionally, in Figure 4, we run the iterative methods from Table 1 (IRKSN, IRCR, IROSR and SRDI) (as well as IHT for comparison) on Example 1, and measure the recovery error $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^{(y)}\|$ as well as the sparsity $\|\hat{\boldsymbol{w}}\|_0$ of the iterates. As we can see, only IRKSN can achieve 0 error, that is, full recovery in the noiseless setting. In addition, except IHT (which however fails to approach the true solution), no method is able to converge to a 3-sparse solution, which is the true degree of sparsity of the solution.

Below we present experimental results to evaluate the sparse recovery properties of IRKSN. Additional details on those experiments as well as further experiments are provided in the Appendix.

Experimental Setting. We consider a simple linear regression setting with a correlated design matrix, i.e. where the design matrix $\boldsymbol{X}$ is formed by n i.i.d. samples from d (we take $d = 50$ here) correlated Gaussian random variables $\{X_1, .., X_d\}$ of zero mean and unit variance, such that: $\forall i \in \{1, \ldots, d\} : \mathbb{E}[X_i] = 0, \mathbb{E}[X_i^2] = 1$; and $\forall (i, j) \in$

$\{1, \dots, d\}^2, i \neq j : \mathbb{E}[X_i X_j] = \rho^{|i-j|}$. More precisely, we generate each feature X_i in an auto-regressive manner, from previous features, using a correlation $\rho \in [0, 1)$, in the following way: we have $X_1 \sim \mathcal{N}(0, 1)$ and $\sigma^2 = 1 - \rho^2$, and for all $j \in \{2, ..., d\}$: $X_{j+1} = \rho X_j + \epsilon_j$ where $\epsilon_j = \sigma * \Delta$, with $\Delta \sim \mathcal{N}(0, 1)$. Additionally, $\boldsymbol{w}$ is supported on a support, sampled uniformly at random, of $k = 10$ non-zero entries, with each non-zero entry sampled from a normal distribution, and $\boldsymbol{y}$ is obtained with a noise vector $\boldsymbol{\epsilon}$ created from i.i.d. samples from a normal distribution, rescaled to enforce a given signal to noise ratio (SNR), as follows: $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\epsilon}$ with the signal to noise ratio defined as $\text{snr} = \frac{\|\boldsymbol{X}\boldsymbol{w}^*\|}{\|\boldsymbol{\epsilon}\|}$. We generate this dataset using the `make_correlated_data` function from the `benchopt` package (Moreau et al. 2022). Such a dataset is commonly used to evaluate sparse recovery algorithms (see e.g. (Molinari et al. 2021)), since it possesses correlated features, which is more challenging for sparse recovery (see e.g. the ElasticNet paper, which was motivated by such correlated datasets (Zou and Hastie 2005)). In addition, the advantage of such synthetic dataset is that the support is known since it is generated, which therefore allows to evaluate the performance of the algorithms on support recovery, contrary to real-life datasets where a true sparse support of $\boldsymbol{w}$ is hypothetical (or at least often unknown). Additionally, we can notice that such dataset resembles our Example 1, as some features are generated from other features. We evaluate the performance of each final recovered model $\boldsymbol{w}$ using the F1 score on support recovery, defined as follows: $\text{F1} = 2\frac{PR}{P+R}$, with P the precision and R the recall of support recovery, which are defined as: $P = \frac{|\text{supp}(\boldsymbol{w}^*) \cap \text{supp}(\boldsymbol{w})|}{|\text{supp}(\boldsymbol{w})|}$ and $R = \frac{|\text{supp}(\boldsymbol{w}^*) \cap \text{supp}(\boldsymbol{w})|}{|\text{supp}(\boldsymbol{w}^*)|}$. Therefore, the F1 score allows to evaluate at the same time how much of the predicted nonzero elements are accurate, and how much of the actual support has been found. A higher F1 score indicates better identification of the true support. In each experiment (defined by a particular value of $n, \rho,$ snr and a given random seed for generating X, $\boldsymbol{w}^*$ and $\boldsymbol{\epsilon}$), and for each algorithm, we choose the hyperparameters from a grid-search, to attain the best F1-score (we give details on that grid in the Appendix). For all algorithms which need to set a value k (IRKSN, KSN, IHT), we set k to its true value $k = 10$. In a realistic use-case, since the support is unknown, one may instead tune those hyper-parameters based on a hold-out validation set prediction mean squared error, but tuning those hyperparameters directly for best support F1 score, as we do, allows to evaluate the best potential support recovery capability of each algorithm (e.g. for Lasso it informs us that *there exist* a certain λ, such that we can achieve such a support recovery score). Each experiment is regenerated 5 times with different random seeds, and the average of the obtained best F1 scores, as well as their standard deviation, are reported in Figures 5(a), 5(c), and 5(b), for various values of the dataset parameters, while the others are kept fixed. In Figure 5(a), we take $\rho = 0.5$, $\text{snr} = 1.$, and $n \in \{10, 30, 50, 70, 90\}$. In Figure 5(b), we take $\rho = 0.5$, $\text{snr} \in \{0.1, 0.5, 1., 2., 3.\}$, and $n = 30$. In Figure 5(c), we take $\rho \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\text{snr} = 1.$, and $n = 30$. Additionally, we plot on Figure 5(d) the evolution



(a) F1-score vs. n (b) F1-score vs. snr

(c) F1-score vs. ρ (d) F1-score vs. t

Figure 5: F1-score of support recovery in various settings

of the F1 score along training for iterative algorithms (i.e. algorithms where there is no grid search over a penalty λ, which are IHT, IRKSN, IRCR, IROSR, SRDI), in the case where $n = 30, \text{snr} = 3$, and $\rho = 0.5$.

Results. In all the experiments, as can be expected, we observe that support recovery is more successful when the signal to noise ratio is high, the number of samples is greater, and the correlation ρ is smaller (for that latter point, this is due to the fact that highly correlated datasets are harder for sparse recovery, see e.g. (Zou and Hastie 2005) for a discussion on the topic). But overall, we can observe that IRKSN consistently achieves better support recovery than other algorithms from Table 1. Also, we can observe on Figure 5(d) that IHT and IRKSN maintain a good F1 score after many iterations, while other methods implicitly enforcing an ℓ_1 norm regularization (IRCR, IROSR, SRDI) have poor F1 score in late training.

Conclusion

In this paper, we introduced an iterative regularization method based on the k-support norm regularization, IRKSN, to complement usual methods based on the ℓ_1 norm. In particular, we gave some condition for sparse recovery with our method, that we analyzed in details and compared to traditional conditions for recovery with ℓ_1 norm regularizers, through an illustrative example. We then gave an early stopping bound for sparse recovery with IRKSN with explicit constants in terms of the design matrix and the true sparse vector. Finally, we evaluated the applicability of IRKSN on several experiments. In future works, it would be interesting to analyze recovery with the s-support norm for general s, where s is not necessarily equal to k: indeed, this setting would generalize both our work and works based on the ℓ_1 norm. We leave this for future work.

Acknowledgements

We would like to thank Velibor Bojković for fruiteful discussions, as well as the anonymous reviewers for their useful comments. Xiao-Tong Yuan is funded in part by the National Key Research and Development Program of China under Grant No. 2018AAA0100400, and in part by the Natural Science Foundation of China (NSFC) under Grant No.U21B2049 and No.61936005.

References

Argyriou, A.; Foygel, R.; and Srebro, N. 2012. Sparse prediction with the k-support norm. *Advances in Neural Information Processing Systems*, 25.

Belilovsky, E.; Gkirtzou, K.; Misyrlis, M.; Konova, A. B.; Honorio, J.; Alia-Klein, N.; Goldstein, R. Z.; Samaras, D.; and Blaschko, M. B. 2015. Predictive sparse modeling of fMRI data for improved classification, regression, and visualization using the k-support norm. *Computerized Medical Imaging and Graphics*, 46: 40–46.

Bertrand, Q.; and Massias, M. 2021. Anderson acceleration of coordinate descent. In *International Conference on Artificial Intelligence and Statistics*, 1288–1296. PMLR.

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

Bubeck, S.; et al. 2015. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4): 231–357.

Cai, J.-F.; Osher, S.; and Shen, Z. 2009. Linearized Bregman iterations for compressed sensing. *Mathematics of computation*, 78(267): 1515–1536.

Chatterjee, S.; Chen, S.; and Banerjee, A. 2014. Generalized dantzig selector: Application to the k-support norm. *Advances in Neural Information Processing Systems*, 27.

Fang, H.; Fan, Z.; Sun, Y.; and Friedlander, M. 2020. Greed meets sparsity: Understanding and improving greedy coordinate descent for sparse optimization. In *International Conference on Artificial Intelligence and Statistics*, 434–444. PMLR.

Foucart, S.; and Rauhut, H. 2013. An invitation to compressive sensing. In *A mathematical introduction to compressive sensing*, 1–39. Springer.

Gkirtzou, K.; Honorio, J.; Samaras, D.; Goldstein, R.; and Blaschko, M. B. 2013. fMRI analysis of cocaine addiction using k-support sparsity. In *2013 IEEE 10th International Symposium on Biomedical Imaging*, 1078–1081. IEEE.

Golub, G. H.; and Van Loan, C. F. 2013. *Matrix computations*. JHU press.

Golub, T. R.; Slonim, D. K.; Tamayo, P.; Huard, C.; Gaasenbeek, M.; Mesirov, J. P.; Coller, H.; Loh, M. L.; Downing, J. R.; Caligiuri, M. A.; et al. 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439): 531–537.

Grasmair, M.; Scherzer, O.; and Haltmeier, M. 2011. Necessary and sufficient conditions for linear convergence of ℓ_1-regularization. *Communications on Pure and Applied Mathematics*, 64(2): 161–182.

Jacob, L.; Obozinski, G.; and Vert, J.-P. 2009. Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning*, 433–440.

Jain, P.; Tewari, A.; and Kar, P. 2014. On Iterative Hard Thresholding Methods for High-dimensional M-Estimation. In *Advances in Neural Information Processing Systems*, volume 27.

Jia, J.; and Yu, B. 2010. On model selection consistency of the elastic net when p ¿¿ n. *Statistica Sinica*, 595–611.

Lang, K. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, 331–339. Elsevier.

Matet, S.; Rosasco, L.; Villa, S.; and Vu, B. L. 2017. Don't relax: early stopping for convex regularization. *arXiv preprint arXiv:1707.05422*.

McDonald, A.; Pontil, M.; and Stamos, D. 2016a. Fitting spectral decay with the k-support norm. In *Artificial Intelligence and Statistics*, 1061–1069. PMLR.

McDonald, A. M.; Pontil, M.; and Stamos, D. 2014. Spectral k-support norm regularization. *Advances in neural information processing systems*, 27.

McDonald, A. M.; Pontil, M.; and Stamos, D. 2016b. New perspectives on k-support and cluster norms. *The Journal of Machine Learning Research*, 17(1): 5376–5413.

Molinari, C.; Massias, M.; Rosasco, L.; and Villa, S. 2021. Iterative regularization for convex regularizers. In *International conference on artificial intelligence and statistics*, 1684–1692. PMLR.

Moreau, T.; Massias, M.; Gramfort, A.; Ablin, P.; Bannier, P.-A.; Charlier, B.; Dagréou, M.; La Tour, T. D.; Durif, G.; Dantas, C. F.; et al. 2022. Benchopt: Reproducible, efficient and collaborative optimization benchmarks. In *NeurIPS-36th Conference on Neural Information Processing Systems*.

Natarajan, B. K. 1995. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2): 227–234.

Nesterov, Y. 2009. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1): 221–259.

Osher, S.; Ruan, F.; Xiong, J.; Yao, Y.; and Yin, W. 2016. Sparse recovery via differential inclusions. *Applied and Computational Harmonic Analysis*, 41(2): 436–469.

Parikh, N.; Boyd, S.; et al. 2014. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3): 127–239.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1): 267–288.

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

Vaskevicius, T.; Kanade, V.; and Rebeschini, P. 2019. Implicit regularization for optimal sparse recovery. *Advances in Neural Information Processing Systems*, 32.

Wang, J.; Kwon, S.; Li, P.; and Shim, B. 2015. Recovery of sparse signals via generalized orthogonal matching pursuit: A new analysis. *IEEE Transactions on Signal Processing*, 64(4): 1076–1089.

Wright, J.; and Ma, Y. 2022. *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications*. Cambridge University Press.

Xiao, L. 2009. Dual Averaging Method for Regularized Stochastic Learning and Online Optimization. In *Advances in Neural Information Processing Systems*, volume 22.

Zhao, P.; Yang, Y.; and He, Q.-C. 2022. High-dimensional linear regression via implicit regularization. *Biometrika*.

Zou, H.; and Hastie, T. 2005. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2): 301–320.