The k-Support Norm and Convex Envelopes of Cardinality and Rank

Anders Eriksson
School of Electrical Engineering
and Computer Science
Queensland University of Technology
anders.eriksson@qut.edu.au

Trung Thanh Pham, Tat-Jun Chin, Ian Reid
School of Computer Science
The University of Adelaide
{trung.pham,tat-jun.chin,ian.reid@adelaide.edu.au}

Abstract

Sparsity, or cardinality, as a tool for feature selection is extremely common in a vast number of current computer vision applications. The k-support norm is a recently proposed norm with the proven property of providing the tightest convex bound on cardinality over the Euclidean norm unit ball. In this paper we present a re-derivation of this norm, with the hope of shedding further light on this particular surrogate function. In addition, we also present a connection between the rank operator, the nuclear norm and the k-support norm. Finally, based on the results established in this re-derivation, we propose a novel algorithm with significantly improved computational efficiency, empirically validated on a number of different problems, using both synthetic and real world data.

1. Introduction

The use of sparsity in the field of Computer Vision can perhaps best be motivated by two main principles, computational efficiency and model simplicity. Selecting as few variables or features as possible will hopefully yield both a cheap and accurate model of the problem at hand.

Sparsity is typically obtained by regularizing the goodness of fit with the cardinality, (denoted card$(\cdot)$ or $||\cdot||_0$) of the variables. However, as this typically leads to non-convex optimization problems with high computational demands, the standard approach is to replace and relax these regularizers with convex surrogates for cardinality. The most popular such surrogate function is unquestionably the l_1-norm. The use of which is regularly justified by saying that the l_1-norm makes up the *convex envelope* of the cardinality function. However, as it was pointed out in [1], this is an argument that must be used with care. Correctly stated, we have that the l_1-norm is only the convex envelope of $||\cdot||_0$ on the bounded domain, $\{x \in \mathbb{R}^d \mid ||x||_\infty \leq 1\}$, the l_∞-norm unit ball.

It was argued in [1] that in certain instances it might be reasonable to not only expect that each individual entry of the entering variables be bounded, but to have bounds on their Euclidean norm as well. This led to the proposal of the k-support norm, a norm that was shown to provide the tightest convex relaxation of cardinality on the Euclidean-norm unit ball. It was also shown that this norm leads to improved learning guarantees as well as better algorithmic stability. However, as [1] did not address computational complexity issues, the proposed algorithm, employing an exhaustive search method, has proven painfully slow for even modest sized problems.

The authors of [10] presented an improved algorithm for computing the solution of k-support norm regularized optimization problems. They proposed the use of binary search as a replacement for certain subproblems in the original algorithm of [1]. Despite reporting significant speed-ups over [1], this method was still exhaustive in nature and could for larger problems and/or certain parameter choices still prove to be quite inefficient.

In this paper we attempt to make progress towards shedding further light on a number of different aspects of the k-support norm. In our opinion, there are three main contributions made here. Firstly, we present a slightly different derivation of the k-support norm with a stronger emphasis on the concept of convex envelopes. By doing so we hope to provide a different perspective to previous work and also make the connection to the rank operator on matrices perhaps more obvious. Secondly, we show that there is an equivalence between cardinality, the rank operator and the nuclear norm on domains shared between the elements of vectors and singular values of matrices. Our final contribution is a proposed algorithm for solving optimization problems involving the k-support norm. This method is then empirically proven to be orders of magnitude faster than the existing state-of-the-art approaches.

2. The k-support Norm and Convex Envelopes of Cardinality

We start by presenting some prerequisites needed to establish the main results of this paper. This includes a re-derivation of the the k-support norm, as defined in [1]. We do this in an attempt to further the understanding and intuition behind this norm as well as to make the connection between convex envelopes of cardinality and rank over different domains more clear.

First let us consider the set,

$$\mathcal{C}_k^{(\infty)} = \left\{x \in \mathbb{R}^d \;\middle|\; ||x||_0 \leq k,\ ||x||_\infty \leq 1\right\}. \tag{1}$$

It is a well known result that the convex hull of (1) is given by

$$\text{conv}(\mathcal{C}_k^{(\infty)}) = \left\{x \in \mathbb{R}^d \;\middle|\; ||x||_1 \leq k,\ ||x||_\infty \leq 1\right\}. \tag{2}$$

This result can be realised as the Fenchel biconjugate of the indicator function of $\mathcal{C}_k^\infty$,

$$\chi_{\mathcal{C}}(x) = \begin{cases} 0, & x \in \mathcal{C} \\ \infty, & x \notin \mathcal{C} \end{cases} \tag{3}$$

From its definition we first obtain the Fenchel conjugate f_∞^* as

$$f_\infty^*(y) = \sup_x y^T x - \chi_{\mathcal{C}_k^\infty}(x) = \sum_{i=1}^k |y|_i^\downarrow, \tag{4}$$

where $|y|_i^\downarrow$ denotes the i-th largest element, by magnitude, of the vector $y \in \mathbb{R}^d$. The biconjugate f_∞^{**} is given by

$$f_\infty^{**}(x) = \sup_y x^T y - \sum_{i=1}^k |y|_i^\downarrow =$$
$$= \begin{cases} 0, & ||x||_\infty \leq 1,\ \frac{1}{k}||x||_1 \leq 1 \\ \infty, & \text{otherwise} \end{cases} = \chi_{\text{conv}(\mathcal{C}_k^\infty)}(x), \tag{5}$$

effectively verifying (2). For the last equality in (5) see [2, Sec.IV 1.18]. Since $k \geq 1$ and here $k||x||_\infty \geq ||x||_1$ it follows from (5) that, on the domain $||x||_\infty \leq 1$, (the unit ball of the l_∞-norm), the convex envelope of the cardinality function $||\cdot||_0$ is the l_1-norm. This result is then frequently taken as motivation to use the l_1-norm as a surrogate for cardinality. However, as pointed out in [1], this result is sometimes used without the appropriate care, omitting the domain $||x||_\infty \leq 1$ for which this convex envelope is valid. Over different domains the convex envelope of cardinality may in fact be distinctly different.

It was perhaps this insight that in part led to the original derivation of the k-support norm in [1]. In this work they argue that the l_2-norm might be a more likely and appropriate bound on the entering vectors in certain instances. They therefore instead consider the set

$$\mathcal{C}_k^{(2)} = \left\{x \in \mathbb{R}^d \;\middle|\; ||x||_0 \leq k,\ ||x||_2 \leq 1\right\}, \tag{6}$$

and proceed to show that the convex envelope of this set on the ball $||x||_2 \leq 1$ is given by a norm they denote the k-support norm. The k-support norm $||\cdot||_k^{sp}$ is defined as the gauge function with its unit ball coinciding with the convex hull of $\mathcal{C}_k^{(2)}$.

An alternate way of deriving this convex envelope is by proceeding as above and finding the Fenchel biconjugate of $\chi_{\mathcal{C}_k^{(2)}}$. We obtain

$$f^*(y) = \sup_x y^T x - \chi_{\mathcal{C}_k^{(2)}}(x) = \sup_{\substack{||x||_0 \leq k \\ ||x||_2 \leq 1}} y^T x = \tag{7}$$

$$= \sqrt{\sum_{i=1}^k (|y|_i^\downarrow)^2} = ||y||_k^{(2)}, \tag{8}$$

where $||\cdot||_k^{(p)} : \mathbb{R}^n \mapsto \mathbb{R}$ is the vector k-norm, defined as the l_p-norm of the k largest component values, in magnitude, of any vector in $\mathbb{R}^d$, also known as a symmetric gauge norm. Next,

$$f^{**}(x) = \sup_y x^T y - ||y||_k^{(2)} = \chi_{||x||_k^{(2)*} \leq 1}(x). \tag{9}$$

Here we used the property that the Fenchel conjugate of a norm is the indicator function of the unit ball of its dual norm. The convex envelope of the cardinality function $||\cdot||_0$ over $||x||_2 \leq 1$ is thus given by the dual norm of vector k-norm $||\cdot||_k^{(2)}$. In [1] it was shown that $||\cdot||_k^{(2)*}$ (the dual norm of $||\cdot||_k^{(2)}$) is indeed the k-support norm as defined therein and consequently it follows that $||\cdot||_k^{(2)}$, again, must be the convex envelope of the cardinality function on the unit ball $||x||_2 \leq 1$.

However, we will show that the dual norm $||\cdot||_k^{(2)}$ is significantly more convenient to work with, rather than the actual k-support norm. An explicit formula for calculating this primal norm is in fact given in [1] but this expression is obtained indirectly through the dual norm by solving

$$\frac{1}{2}(||x||_k^{sp})^2 = \min_y x^T y - \frac{1}{2}(||y||_k^{(2)})^2 \tag{10}$$

This equality follows from an identity that holds for any norm $||\cdot||$, that $\frac{1}{2}||x||^2 = \sup_y x^T y - \frac{1}{2}||y||^{*2}$. The above problem can be reformulated as a linearly constrained quadratic program, (see Prop 2.1 [1]). The resulting formula is then a consequence of the necessary and sufficient conditions for optimality of this reformulation. The integer r of that proposition is in fact directly related to the indices of

the non-zero dual variables of (10). We will return to this relationship between existing formulations and ones given here, in the following sections. We will also in this paper attempt to show that carrying out the calculations entirely in the dual space will not only result in more manageable formulations but will also lead to insights useful in arriving at our proposed algorithm, a more efficient method for minimizing k-support norm regularized loss functions.

3. Sparsity Regularized Parameter Estimation

It was proposed in [1] that the k-support norm would be an appropriate regularizer for general learning problems. In particular when constraints on sparsity as well as the l_2-norm are present. The general form of the problem considered in the original paper is as follows,

$$\min_{x\in\mathcal{D}} \; l(x) + \frac{\gamma}{2}(||x||_k^{sp})^2. \tag{11}$$

Where $k \in [1, d]$ and $\gamma > 0$ is a regularization parameter. The function $l : \mathcal{D} \mapsto [-\infty, +\infty]$ is a smooth, convex loss-function with a Lipschitz continuous gradient and $\mathcal{D} \subseteq \mathbb{R}^d$ a convex domain.

The problem (11) can then efficiently be solved using first-order accelerated proximal gradient methods [4]. This class of methods require the computation of the gradient ∇l and the proximity operator of the spectral norm. For non-smooth loss functions different proximal methods could instead be considered, such as Douglas-Rachford Splitting where $l \in C^1$ is not a requirement, resulting in similar algorithms. However, such methods will not be dealt within this paper.

Definition 3.1 *The proximity operator,* $\text{prox}_f : \mathbb{R}^d \mapsto \mathbb{R}^d$ *is defined as*

$$\text{prox}_f(y) = \arg\min_x \left[f(x) + \frac{1}{2}||x - y||_2^2 \right] \tag{12}$$

where $f : \mathbb{R}^d \mapsto \mathbb{R}$ *is convex and lower semi-continuous.*

Here we will particularly address the issue of efficiently evaluating proximity operators originating from problems on the form (11).

3.1. The Proximity Operator of the k-support Norm

Next we will establish a number of properties of the proximity operator of the square of the k-support norm necessary for the derivation and analysis of our proposed method.

With $f(x) = ||x||_k^{sp}$ the proximity operator related to (11) will have the form

$$\text{prox}_{\gamma/2f^2}(v) = \arg\min_{x\in\mathbb{R}^d} \left[\frac{1}{2}||x - v||_2^2 + \frac{\gamma}{2}(||x||_k^{sp})^2 \right], \tag{13}$$

where v is obtained from the gradient of l. In [1, 10] this operator was solved using exhaustive search and a combination of exhaustive search and binary search respectively.

From the Moreau decomposition [13, 14] we have that for a proper, convex and lower semicontinuous function $f : \mathbb{R}^d \mapsto \mathbb{R}$ the following relationship holds:

$$v = \text{prox}_f(v) + \text{prox}_{f^*}(v). \tag{14}$$

That is, any proximity operator can be computed via the proximity operator of its Fenchel conjugate. We can now write,

$$\text{prox}_{(\gamma/2f^2)^*}(v) = \arg\min_{y\in\mathbb{R}^d} \left[\frac{1}{2}||y - v||_2^2 + \frac{1}{2\gamma}(||y||_k^{sp*})^2 \right] \tag{15}$$

Here we used the same property as in (10). We argue that this dual proximity operator, at least for our purposes, is a more convenient representation for the problem (13).

Proposition 3.1 *For* $y = \text{prox}_{(\gamma/2f^2)^*}(v)$*, the following properties hold*

(a) Elementwise non-expansive; $|y_i| \leq |v_i|, \; \forall i \in [1, ..., d]$.

(b) Sign-preserving; $\text{sign}(v_i) = \text{sign}(y_i), \; \forall i \in [1, ..., d]$.

(c) Order-preserving; if $|v_i| \geq |v_j|$ *for some* $i, j \in [1, ..., d]$*. then* $|y_i| \geq |y_j|$

As a consequence of proposition 3.1 we can then without loss of generality assume that v is non-negative and sorted, so $v_i \geq v_{i+1} \geq 0, \; i = 1, ..., d-1$. Then (15) is equivalent to the following quadratic optimization problem,

$$\begin{aligned} \arg\min_{\substack{y\in\mathbb{R}^d \\ y_i \geq y_{i+1}, \\ i=1,...,d-1}} \quad & h(y) = y^T E^\gamma y - 2\gamma v^T y. \end{aligned} \tag{16}$$

Here $E^\gamma \in \mathbb{R}^{d\times d}$ a positive definite, diagonal matrix with entries $[E^\gamma]_{ij} = \begin{cases} 1+\gamma, & i=j, \; i\leq k \\ \gamma, & i=j, \; i>k \\ 0, & i\neq j \end{cases}$.

Definition 3.2 *Given a constrained convex optimization problem, with the feasible region defined by a set of functions* $g_1(x) \leq 0, ..., g_l(x) \leq 0$*. A constraint* $g_i(x) \leq 0$ *is defined as* ***inactive*** *if it can be removed without influencing the final result of the optimization. Constraints that are not inactive are defined as* ***active****.*

Let $\mathcal{I}^* \subseteq \{1, ..., d-1\}$ denote the set of active constraints at the optima of (16).

Proposition 3.2 *The set of active constraints* $\mathcal{I}^*$ *at optimality of* (16) *is either empty or an interval containing* k*, i.e* $\mathcal{I}^* = [\kappa^l, \; \kappa^u]$, $\kappa^l \leq k \leq \kappa^u$.

If we know $\mathcal{I}^* = [\kappa^l,\ \kappa^u]$ then finding the minimizer y^* of (16) is trivial and given by,

$$y_i^* = \begin{cases} \frac{\gamma v_i}{1+\gamma}, & i < \kappa^l \\ \frac{\gamma \sum_{j\in\mathcal{I}^*} v_j}{\sum_{j\in\mathcal{I}^*}[E^\gamma]_{jj}}, & i \in \mathcal{I}^* \\ v_i, & i > \kappa^u \end{cases} \quad (17)$$

3.2. Bounds on γ

Before we state our proposed algorithm for solving (13) we briefly establish some further relationships between the dual formulation of the previous section and that of [1].

There the k-support norm was interpreted as a trade-off between the l_2 and l_1 norms. This interpretation can also be realised by noting that for a given $\mathcal{I}^*$ then (16) is equivalent to

$$\text{prox}_{(\gamma/2f^2)^*}(v) = \arg\min_{y\in\mathbb{R}^d} \Big[\frac{1}{2}||y-v||_2^2 + \frac{1}{2\gamma}||y_{1:\kappa^l-1}||_2^2 + \frac{(k-\kappa^l+1)}{2\gamma}||y_{\kappa^l:d}||_\infty^2\Big]. \quad (18)$$

The dual problem of which is

$$\min_{x\in\mathbb{R}^d} \Big[\frac{1}{2}||x-v||_2^2 + \frac{\gamma}{2}||x_{1:\kappa^l-1}||_2^2 + \frac{\gamma}{2(k-\kappa^l+1)}||x_{\kappa^l:d}||_1^2\Big] \quad (19)$$

and the aforementioned norm trade-off becomes apparent.

A further observation that can be made is that if either $\kappa^l = 1$ or $\mathcal{I}^* = \emptyset$ the k-support norm reduces to the l_1-norm or the vector k-norm respectively both of which have closed form solutions. We can in fact establish bounds on γ for when either of these cases occur by the following proposition.

Proposition 3.3 *For the primal proximity operator* (13) *we have the following bounds on* γ.

(a) If $\gamma \geq \frac{v_{k+1}}{v_k - v_{k+1}}$ *then the squared* k*-support regularization term of* (13) *reduces to* $(||\cdot||_k^{(2)})^2$.

(b) If $\gamma \leq \frac{(k-1)v_1}{\sum_{i=1}^d v_i - dv_i}$ *then the squared* k*-support regularization term of* (13) *reduces to* $||\cdot||_1^2$.

4. Convex Envelopes of Rank

In this section we extend the notion of the k-support norms to matrices and show that it is related to the convex envelope of the rank operator over a particular domain. The connection between cardinality and rank is well known, see for instance [15], and is perhaps best realised from the fact that for a diagonal matrix, constraints on its rank directly equate to constraints on the cardinality of its diagonal elements. For general matrices this can be shown to translate as the cardinality of its singular values. It is then a natural extension to apply convex relaxations of cardinality and sparsifying vector norms as a surrogate for rank.

One of the more popular such surrogates is perhaps the nuclear norm, denoted[1] $||\cdot||_*$, also known as the trace norm or Ky-Fan norm. It was in [7] proven that the convex envelope of the rank function on the domain of the matrix operator norm unit ball is indeed the nuclear norm. This result can be obtained by showing that for $f(\cdot) = \text{rank}(\cdot) : \mathbb{R}^{m\times n} \mapsto \mathbb{N}^+$ its Fenchel biconjugate becomes

$$f^{**}(X) = ||X||_*,\ X \in \{X \in \mathbb{R}^{m\times n} \mid ||X|| \leq 1\},\ (20)$$

cf. (5).

With a similar argument to that of section 2, obtaining convex envelopes of rank on different domains can then be attempted. On the domain of the Frobenius norm unit ball we have the following result.

Theorem 4.1 *The convex envelope of the indicator function* $\chi_{\mathcal{D}_k^{(F)}}$ *of the set* $\mathcal{D}_k^{(F)} = \{X \in \mathbb{R}^{m\times n} \mid rank(X) \leq k,\ ||X||_F \leq 1\}$, *becomes*

$$f^{**}(X) = \chi_{||X||_{*k}^{sp}}. \quad (21)$$

Where $||X||_{*k}^{sp}$ *denotes a spectral* k*-support norm. Let* σ *denote the vector of* $\min(m,n)$ *singular values of* X, *then the spectral* k*-support norm is given by*

$$||X||_{*k}^{sp} = ||\sigma||_k^{sp}. \quad (22)$$

An interesting intuitive verification of the above result might be reached with the following observation. On the domain of the matrix operator norm unit ball the convex envelope of rank is given by the nuclear norm. Similarly, on the domain of the matrix Frobenius norm unit ball the convex envelope of rank is given by the spectral k-support norm. These relationships and more are summarised in table 1. This table further illustrates the equivalence between cardinality and rank pointed out in [15] and their relation to the k-support norm.

5. Efficiently Solving the Proximity Operator for the k-support Norm

The proposed method for solving the proximity operator associated with the k-support norm is presented and summarised in this section.

Our algorithm is motivated by the results of proposition 3.2 and (17). The underlying idea is that if we can find the set of active constraints then solving $\text{prox}_{\gamma/2f^2}(v)$ is straightforward.

[1] Not to be confused with the dual norm $||\cdot||^*$.

Concept:	cardinality	
Elements:	vectors, $x \in \mathbb{R}^d$	
Domain:	$\|\|x\|\|_\infty \leq 1$	$\|\|x\|\|_2 \leq 1$
Convex Surrogate:	$\|\|x\|\|_1$	$\|\|x\|\|_k^{sp}$
Concept:	rank	
Elements:	matrices, $X \in \mathbb{R}^{m \times n}$	
Domain:	$\|\|X\|\| = \|\|\sigma\|\|_\infty \leq 1$	$\|\|X\|\|_F = \|\|\sigma\|\|_2 \leq 1$
Convex Surrogate:	$\|\|X\|\|_* = \|\|\sigma\|\|_1$	$\|\|X\|\|_{*k}^{sp} = \|\|\sigma\|\|_k^{sp}$

Table 1. Summary of the convex surrogates and different domains discussed in this paper.

Let $\mathcal{I} = [\kappa^l, \kappa^u]$ be a subinterval of $\{1, .., d\}$ containing k. The set $\mathcal{I}$ is called *valid* if

$$v_{\kappa^l} < h_\gamma^l(\kappa^l, \kappa^u) = \frac{(1+\gamma)\sum_{j=\kappa^l}^{\kappa^u} v_j}{\sum_{j=\kappa^l}^{\kappa^u}[E^\gamma]_{jj}} \tag{23}$$

$$v_{\kappa^u} > h_\gamma^u(\kappa^l, \kappa^u) = \frac{\gamma\sum_{j=\kappa^l}^{\kappa^u} v_j}{\sum_{j=\kappa^l}^{\kappa^u}[E^\gamma]_{jj}}. \tag{24}$$

This follows directly from applying (17) to definition 3.2.

Proposition 5.1 *Let $\mathcal{I}$ be a valid subset.*

(a) If $v_{\kappa^l - 1} < h_\gamma^l(\kappa^l - 1, \kappa^u)$ then $\mathcal{I} \cup \{\kappa^l - 1\}$ is also valid.

(b) If $v_{\kappa^u + 1} > h_\gamma^u(\kappa^l, \kappa^u + 1)$ then $\mathcal{I} \cup \{\kappa^u + 1\}$ is also valid.

If a valid set $\mathcal{I}$ can not be expanded in accordance with proposition 5.1 we call the subset *maximal*.

Proposition 5.2 *If $\mathcal{I}$ is valid and maximal then $\mathcal{I}^* = \mathcal{I}$.*

The above results suggest a manner of solving (13), by finding a sequence of valid, strictly inclusive subsets $\{\mathcal{I}_t\}_{t=0}^n$, where $\mathcal{I}_0 \subset \mathcal{I}_1 \subset ... \subset \mathcal{I}_n$. If $\mathcal{I}_t$ is valid, we are guaranteed by proposition 5.1 to find a strictly including subset $\mathcal{I}_{t+1}$. Assuming $\mathcal{I}^*$ is non-empty, then with $\mathcal{I}_0 = \{k\}$ (the smallest valid subset) such an approach guaranteed to terminate in a finite number of steps and hence by proposition 5.2 it will converge to the optimal subset $\mathcal{I}^*$.

Unfortunately, increasing the intervals one element at a time will not be sufficiently efficient. However, with the following proposition the computational requirements of our method can be significantly reduced.

Proposition 5.3 *The expressions $h_\gamma^u(\kappa^l, \kappa^u) - v_{\kappa^l} = \frac{(1+\gamma)\sum_{j=\kappa^l}^{\kappa^u} v_j}{\sum_{j=\kappa^l}^{\kappa^u}[E^\gamma]_{jj}} - v_{\kappa^l}$ and $h_\gamma^u(\kappa^l, \kappa^u) - v_{\kappa^u} = \frac{\gamma\sum_{j=\kappa^l}^{\kappa^u} v_j}{\sum_{j=\kappa^l}^{\kappa^u}[E^\gamma]_{jj}}$. are both monotonic in κ^l and κ^u respectively.*

The above proposition permits the use of any dichotomic divide and conquer search algorithm methods for finding the sought after interval. Instead of updating the sequence of intervals $\mathcal{I}_t$ incrementally as above we can instead use for instance binary search algorithm [5] for updating κ^l and κ^u. It can easily be shown that this modification will not affect the convergence properties established earlier. The proposed algorithm is summarized in alg. 1.

Algorithm 1 The proposed algorithm for Solving the proximity operator for the k-support norm.

Input: v, γ
Output: $x = \text{prox}_{(\gamma/2f^2)}(v)$ (solution of (13))
$v' \leftarrow \text{sort}(|v|)$
if $v'_k \geq \frac{1+\gamma}{\gamma} v'_{k+1}$ **then**
  $\mathcal{I}^* \leftarrow \emptyset$ and goto 14
end if
Initialize: $\mathcal{I}_0 \leftarrow [k,\ k]$, $t \leftarrow 0$
repeat
  $\kappa_{t+1}^l \leftarrow \text{BinarySearch}(\kappa^l, 1, \kappa_t^l, h_\gamma^l(\kappa^l, \kappa_t^u) - v_{\kappa^l})$
  $\kappa_{t+1}^u \leftarrow \text{BinarySearch}(\kappa^u, \kappa_t^u, d, v_{\kappa^u} - h_\gamma^u(\kappa_{t+1}^l, \kappa^u))$
  $\mathcal{I}_{t+1} \leftarrow \left[\kappa_{t+1}^l, \kappa_{t+1}^u\right]$
  $t \leftarrow t + 1$
until convergence
Solve for y' given $\mathcal{I}^* = \mathcal{I}_t$ according to (17)
Reorder and impose signs on y' in accordance to v and 3 to obtain y.
$x \leftarrow v - y$

Remark: Here $\text{BinarySearch}(i, u, l, f(i))$ denotes a binary search over i in the interval $[u, l]$ for the smallest i such that $f(i) \leq 0$.

Solving $\text{prox}_{(\gamma/2f^2)}(v)$ can hence be done in $\mathcal{O}(d\ \log d)$ time using algorithm 1. The sorting of $v \in \mathbb{R}^d$ takes $\mathcal{O}(d\ \log d)$. At most k and $d - k$ binary searches are conducted on κ^l and κ^u, each with a complexity of $\mathcal{O}(\log k)$ and $\mathcal{O}(\log(d - k))$, respectively.

6. Experimental Validation

Below we present the numerical results that illustrate the computational performance of the proposed method and formulations. We compared our algorithm with the current state of the art for solving k-support regularized problems. These two methods were both evaluated on a set of synthetic and real world problems. As this work focussed solely on computational efficiency we limit our reporting to computational metrics only. For results on evaluating the performance of the k-support norm as an efficient regularizer we instead refer the readers to the works of [1, 10]. We also present some preliminary results on the spectral k-support norm regularization.

6.1. Synthetic Problems

We first compared computational performance of our method (algorithm 1) and that of Lai, et. al. ([10]) on solving the proximity operator (13). These two algorithms were evaluated on a large number of synthetically generated data.

The vector $v \in \mathbb{R}^d$ was drawn from a uniform distribution, $v = U([0,1])^d$, the dimension was set to $d = 10^5$, the sparsity to $k = 0.1d$ and the regularization term to $\gamma = 10$, unless otherwise stated. The results can be seen in figure 1.

As the two methods considered both assume that the input is sorted according to magnitude, this is an overhead that is present and constant in both methods. We therefore report our computational comparison both including and excluding the time required by the sorting operator. The results are still quite overwhelming in favor of algorithm 1 over that of Lai et. al. Excluding sorting, our approach is up to 500 times faster. When taking overhead of sorting into account, this resulting speed-up drops, as expected, but our algorithm is still orders of magnitude more efficient than [10].

6.2. Real-World Problems

Here we show a computational comparison between the same two algorithms as above but on two real-world applications and data sets. The reported computational requirement is now the total time taken to solve each problem in question, not only solving the proximity operators. As the computational overhead is identical for both methods the speed improvements observed in the previous section will now be slightly smaller yet, in our opinion, still convincing.

6.3. Subspace Clustering

The first problem considered is that of subspace segmentation, see [6]. Given $X \in \mathbb{R}^{m \times n}$, containing n m-dimensional data samples we have a problem on the following form,

$$\min_{W \in \mathbb{R}^{n \times n}} \frac{1}{2}||X - XW||_2^2 + \frac{\gamma}{2}||\text{vec}(W)||_k^{sp}. \qquad (25)$$

Closely following the protocol described in [10] we conducted two experiments, one on Face clustering using the Extended Yale B database [8] and the other on motion segmentation using the Hopkins 155 datasets [16]. The runtime comparisons are shown in tables 2 and 3. It can there be seen that our proposed algorithm is on these datasets as much as 33 times faster than that of Lai et. al.

6.4. Sparse Coding

In this section we considered sparse coding for image classification, with problems on the following form,

$$\min_{W \in \mathbb{R}^{d \times n}} \frac{1}{2}||X - DW||_2^2 + \frac{\gamma}{2}||\text{vec}(W)||_k^{sp}. \qquad (26)$$

Figure 2. Image noise removal using the spectral k-support norm and the nuclear norm. **Left:** Original image. **Middle:** Noisy image. **Right:** Recovered image using the spectral k-support norm.

Here $X \in \mathbb{R}^{m \times n}$ is a matrix of n individual m-dimensional image descriptors. The dictionary matrix $D \in \mathbb{R}^{m \times d}$ consists of d basis vectors, each of dimension m. The vectorization operator, denoted vec : $\mathbb{R}^{m \times n} \mapsto \mathbb{R}^{mn}$, maps matrices onto vectors by stacking the columns of the given matrix on top of one another.

Again following the protocol of [10] we evaluated the run-times of the two methods on 2 separate datasets, the UIUC-Sport dataset [12] and the Scene15 dataset [11]. The results can be seen in tables 4 and 5.

6.5. Spectral k-support Norm

As a final experiment we present some results obtained using the spectral k-support norm for image noise removal. Partially corrupted images can be approximately recovered by viewing the image as a low-rank matrix and applying matrix completion approaches to restore the corrupted pixels. This assumption of low-rank does not in general hold for natural images, however its top singular values will contain a majority of the information contained in it.

We evaluated the performance of the spectral k-support norm and nuclear norms as surrogates for rank on 3 different images. We followed the methodology of [9] closely and randomly corrupted half the pixels of each image. The images were then recovered according to the above procedure. The results are shown in fig 2.

Here we used $k = 20$ and tuned the regularization parameters individually for each norm for the best results. The recovery error is listed in table 6. The actual minimization

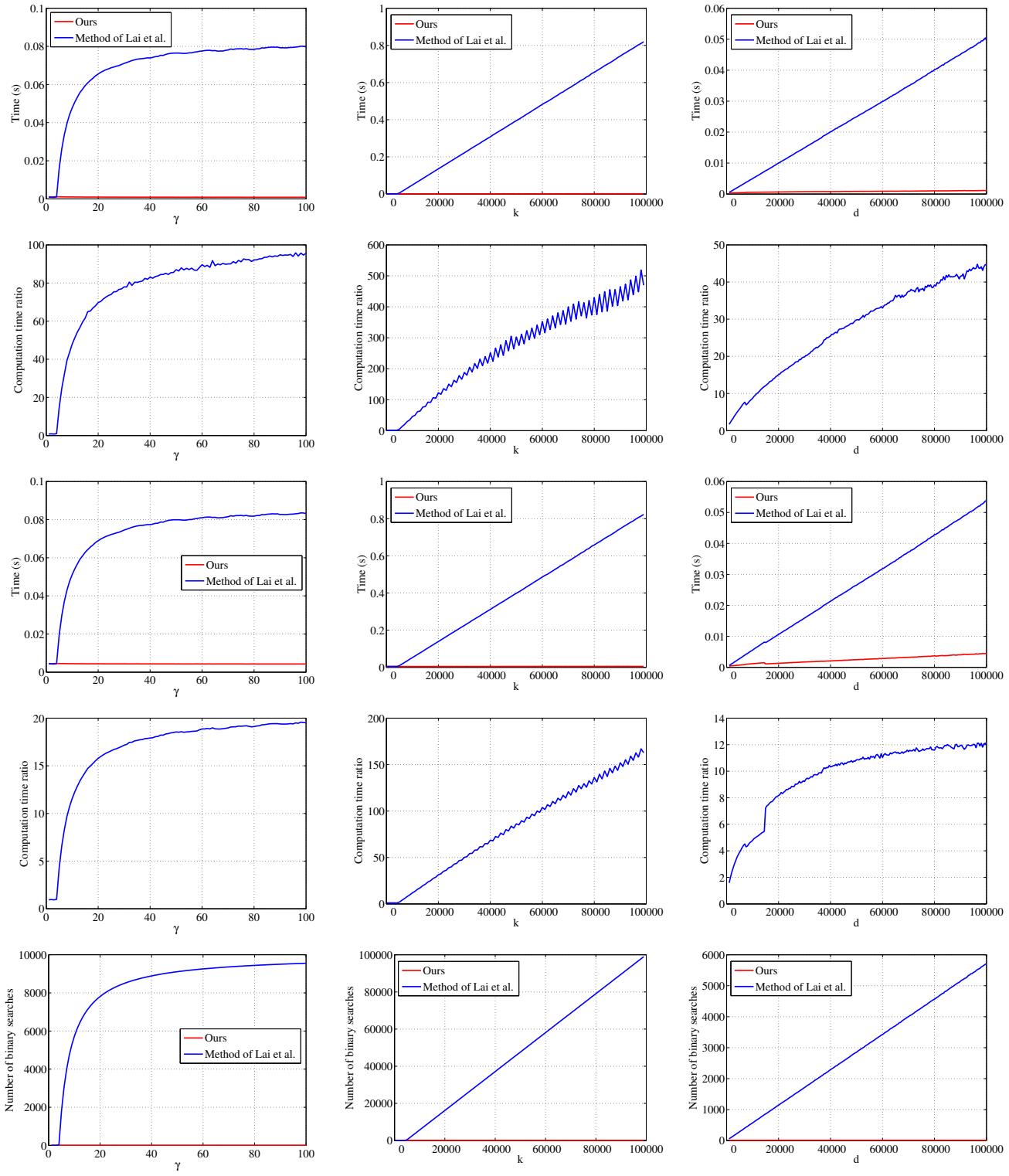

Figure 1. **Row 1:** Computational performance (excl. sorting). **Row 2:** Ratio of speedup (excl. sorting). **Row 3:** Computational performance (incl. sorting). **Row 4:** Ratio of speedup (incl. sorting). **Row 5:** Number of binary searches conducted. In this set of experiments our method typically only required $7 - 8$ binary searches. Each result is the average of 100 runs.

k	$0.005n^2$	$0.01n^2$	$0.05n^2$	$0.1n^2$	$0.2n^2$	$0.3n^2$	$0.4n^2$	$0.5n^2$
Lai et al. (5)	19.90	21.53	41.55	68.53	131.95	201.59	274.47	357.10
Ours (5)	**10.61**	**10.78**	**11.02**	**10.89**	**11.09**	**11.43**	**11.43**	**11.38**
Lai et al. (10)	82.45	89.38	144.30	220.08	393.34	586.28	803.24	1044.99
Ours (10)	**52.18**	**52.98**	**54.22**	**54.97**	**56.08**	**56.27**	**57.48**	**58.10**

Table 2. Running time comparison on face clustering task using the images of the first 5 and 10 classes of the Extended Yale B dataset. The parameter γ is chosen from the set $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$, and the parameter k is selected from $\{0.005n^2, 0.01n^2, 0.05n^2, 0.1n^2, 0.2n^2, 0.3n^2, 0.4n^2, 0.5n^2\}$, where n is the number of images. The table shows the computation time (in sec) under different k, after being averaged over γ.

γ	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^{-0}	10^{1}	10^{2}
Lai et al.	186.24	186.60	189.91	216.52	296.34	369.27	400.32	422.75
Ours	**13.38**	**13.38**	**13.40**	**13.36**	**13.30**	**12.87**	**12.77**	**12.58**

Table 3. Running time comparison on motion segmentation task using the Hopkins 155 datasets. We fix $k = 0.5n^2$ (n is the number of trajectories in each sequence), and vary the values of γ. For each method and value of γ we report the average running time computation over 155 sequences.

Parameters	k	10	20	30	40	50	100
$\gamma = 1$	Lai et al.	**168.43**	**176.91**	**184.89**	**196.31**	208.47	259.22
$\gamma = 1$	Ours	192.53	195.75	198.38	199.36	**200.17**	**204.50**
$\gamma = 10$	Lai et al.	**179.80**	194.49	209.49	220.04	234.89	306.15
$\gamma = 10$	Ours	186.19	**192.64**	**194.08**	**197.25**	**198.85**	**202.67**

Table 4. Running time comparison on sparse coding task using UIUC-Sport dataset. The average run-times of the first 10 images are shown.

Parameters	k	10	20	30	40	50	100
$\gamma = 1$	Lai et al.	**155.36**	**164.83**	**175.77**	**186.31**	197.72	250.53
$\gamma = 1$	Ours	181.70	184.01	184.91	187.14	**188.45**	**193.04**
$\gamma = 10$	Lai et al.	**164.59**	178.91	191.84	206.10	218.67	290.40
$\gamma = 10$	Ours	167.77	**172.98**	**177.66**	**176.63**	**180.33**	**193.95**

Table 5. Running time comparison on sparse coding task using Scene15 dataset. We report the average running times of the first 10 images.

of the problem at hand was obtained by direct application of algorithm 1 to the singular value thresholding method of [3].

Image \ Regularizer	$\|\|\cdot\|\|^{sp}_{*k}$	$\|\|\cdot\|\|_*$
Boat	**1.43**	1.48
Cameraman	**1.21**	**1.21**
Lenna	**1.89**	1.97

Table 6. Comparison of achieved average reconstruction error for the three test images using the nuclear norm and spectral k-support norm.

Although this is in a very limited experimental setting the results do seem to indicate that the k-support norm appear to give slightly more accurate reconstructions. Though promising, these preliminary results would need to be verified in a more extensive experimental evaluation.

7. Conclusions

In this paper we have presented an alternative derivation of the k-support norm that will hopefully help to increase the general understanding of convex envelopes of cardinality. We have also established a connection between the k-support norm and the rank operator on matrices over certain domains. All in agreement with existing and established theory. Finally, perhaps our main contribution was the proposal of a novel algorithm for solving optimization problems regularized by the k-support norm. The behaviour and performance of this method was thoroughly analysed and empirically validated with very convincing results.

Acknowledgements

This research was supported by the Australian Research Council through the Discovery Early Career Researcher Award project DE130101775, the ARC Centre of Excellence in Robotic Vision CE140100016 and the Laureate Fellowship FL130100102.

References

[1] A. Argyriou, R. Foygel, and N. Srebro. Sparse prediction with the k-support norm. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1457–1465. Curran Associates, Inc., 2012.

[2] R. Bhatia. *Matrix Analysis*, volume 169. Springer, 1997.

[3] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20(4):1956–1982, Mar. 2010.

[4] P. L. Combettes and J.-C. Pesquet. Proximal Splitting Methods in Signal Processing. In H. Bauschke, R. Burachik, P. Combettes, V. Elser, D. Luke, and H. E. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011.

[5] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

[6] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *CoRR*, abs/1203.1005, 2012.

[7] M. Fazel. *Matrix Rank Minimization with Applications*. PhD thesis, Stanford University.

[8] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[9] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2117–2130, 2013.

[10] H. Lai, Y. Pan, C. Lu, Y. Tang, and S. Yan. Efficient k-support matrix pursuit. In *European Conference on Computer Vision*, 2014.

[11] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society.

[12] L.-J. Li and F.-F. Li. What, where and who? classifying events by scene and object recognition. In *ICCV*, pages 1–8. IEEE, 2007.

[13] J. J. Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *Comptes Rendus de l'Académie des Sciences (Paris), Série A*, 255:2897–2899, 1962.

[14] N. Parikh and S. Boyd. Proximal Algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.

[15] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3):471–501, Aug. 2010.

[16] R. Tron and R. Vidal. A Benchmark for the Comparison of 3-D Motion Segmentation Algorithms. In *Computer Vision and Pattern Recognition, 2007. CVPR IEEE Conference on*. IEEE, June 2007.