Learning with Dynamic Group Sparsity

Junzhou Huang
Rutgers University
110 Frelinghuysen Road
Piscataway, NJ 08854, USA
jzhuang@cs.rutgers.edu

Xiaolei Huang
Lehigh University
19 Memorial Drive West
Bethlehem, PA 18015, USA
xih206@lehigh.edu

Dimitris Metaxas
Rutgers University
110 Frelinghuysen Road
Piscataway, NJ 08854, USA
dnm@cs.rutgers.edu

Abstract

This paper investigates a new learning formulation called dynamic group sparsity. It is a natural extension of the standard sparsity concept in compressive sensing, and is motivated by the observation that in some practical sparse data the nonzero coefficients are often not random but tend to be clustered. Intuitively, better results can be achieved in these cases by reasonably utilizing both clustering and sparsity priors. Motivated by this idea, we have developed a new greedy sparse recovery algorithm, which prunes data residues in the iterative process according to both sparsity and group clustering priors rather than only sparsity as in previous methods. The proposed algorithm can recover stably sparse data with clustering trends using far fewer measurements and computations than current state-of-the-art algorithms with provable guarantees. Moreover, our algorithm can adaptively learn the dynamic group structure and the sparsity number if they are not available in the practical applications. We have applied the algorithm to sparse recovery and background subtraction in videos. Numerous experiments with improved performance over previous methods further validate our theoretical proofs and the effectiveness of the proposed algorithm.

1. Introduction

The compressive sensing (CS) theory has shown that a sparse signal can be recovered from a small number of its linear measurements with high probability [4, 8]. According to CS, a sparse signal $x \in \mathbb{R}^n$ should be recovered from the following linear random projections:

$$y = \Phi x + e, \tag{1}$$

where $y \in \mathbb{R}^m$ is the measurement vector, $\Phi \in \mathbb{R}^{m \times n}$ is the random projection matrix, $m \ll n$, and e is the measurement noise. The CS theory is magnetic as it implies that the signal $x \in \mathbb{R}^n$ can be recovered from only $m = \mathcal{O}(klog(n/k))$ measurements [4] if x is a k-sparse signal, which means that $x \in \mathbb{R}^n$ can be well approximated using $k \ll n$ nonzero coefficients under some linear transform. It directly leads to the potential of cost saving in digit data capturing. Although the encoding in data capturing only involves simple linear projections, signal recovery requires nonlinear algorithms to seek the sparsest signal from the measurements. This problem can be formulated with l^0 minimization:

$$x_0 = argmin\|x\|_0 \quad while \quad \|y - \Phi x\|^2 < \varepsilon \tag{2}$$

where $\|\cdot\|_0$ denotes the l^0-norm that counts the number of nonzero entries and ε is the noise level. This problem is NP-hard. In the general case, no known procedure can correctly find the sparsest solution more efficiently than exhausting all subsets of the entries for x. One key problem in CS is thus to develop efficient recovery algorithms with nearly optimal theoretical performance guarantees.

One class of algorithms tries to seek the sparest solution by performing basis pursuit (BP) based l^1 minimization using linear programming (LP) instead of l^0 minimization [5, 8]. The l_1-magic used a primal log-barrier approach to perform l^1 minimization [4]. A specialized interior-point method is employed to solve large scale problems by using l^1 regularization [14]. Gradient Projection for Sparse Reconstruction (GPSR) is a fast convex relaxation algorithm [11] to approximate the solution. Iterative greedy pursuit is another well-known class of sparse recovery algorithms. The earliest ones include the matching pursuit [15] and orthogonal matching pursuit (OMP) [23]. Their successors include the stagewise OMP (StOMP) [9] and the regularized OMP (ROMP) [19]. While they are much faster than the BP methods, they require more measurements for perfect recovery and lack provable recovery guarantees. To close this gap, the subspace pursuit (SP) [6] and the compressive sampling matching pursuit (CoSaMP) [18] were proposed recently by incorporating backward steps. They have similar theoretical recovery guarantees as that of the BP methods, while their computation complexity is comparable to

those of the greedy pursuit algorithms.

All of these algorithms do not consider sparse data priors other than sparsity. However, in some practical applications, the nonzero sparse coefficients are often not randomly distributed but group-clustered. They tend to cluster into groups although these clustering group structures are dynamic and unpredictable. (For example, the group number/size/location may be unknown.) A few attempts have been made to utilize these group clustering priors for better sparse recovery [1, 13, 24, 22, 26]. For simplicity, all of them assume that the group structures (such as the group number/size/location) are known before recovery. Moreover, they only consider the case where all groups share a common nonzero coefficient support set [1]. These recovery algorithms either do not have explicit bounds on the minimal number of measurements, or lack provable recovery performance guarantees from noise measurements. While their assumption of the block sparsity structure enables better recovery from fewer measurements with less computation, it is not flexible enough to handle some practical sparse data in which the group structures are unknown and only the sparse group-clustering trend is known. Therefore, none of them can handle dynamic group clustering priors, where we do not know the group structure, and only know the sparsity and group clustering trend.

In this paper, we extend the CS theory to efficiently handle data with both sparsity and dynamic group clustering priors. A dynamic group sparsity recovery algorithm is then proposed based on the extended CS theory. It assumes that the dynamic group clustering sparse signals live in a union of subspaces [2] and proposes an approximation algorithm in this union of subspaces to iteratively prune the signal estimations according to both sparsity and group clustering priors. The group clustering trend implies that, if a point lives in the union of subspaces, its neighboring points would also live in this union of subspaces with higher probability, and vice versa. By enforcing this constraints, the degrees of freedom of the sparse signals have been significantly reduced to a narrower union of subspaces. It leads to several advantages: 1) accelerating the signal pruning process; 2) decreasing the minimal number of necessary measurements; and 3)improving robustness to noise and preventing the recovered data from having artifacts. These advantages enable the proposed algorithm to efficiently obtain stable sparse recovery with far fewer measurements than previous algorithms. Finally, we extended the proposed algorithm to adaptively learn the sparsity numbers when they are not exactly known in practical applications.

The remainder of the paper is organized as follows. Section 2 briefly reviews the CS theory. The extended CS theory and the proposed recovery algorithm are detailed in section 3. Section 4 presents the experimental results when applying the proposed algorithm to sparse recovery and background subtraction respectively on both simulated and practical data. We conclude this paper in Section 5.

[1]The support set of sparse data x is defined as the set of indices corresponding to the nonzero entries in x and denoted by $supp(x)$

2. Theory Review

As we know, the decreasingly sorted coefficients of many real signals rapidly decay according to the power law. Thus, these signals can be well approximated or compressed to k-sparse signals although they are not strictly sparse. In CS, the signal capture and compression are integrated into a single process [3, 8]. Thus, we do not capture a sparse signal $x \in \mathbb{R}^n$ directly but rather capture $m < n$ linear measurements $y = \Phi x$ based on a measurement matrix $\Phi \in \mathbb{R}^{m\times n}$. Suppose the set of k-sparse signals $x \in \mathbb{R}^n$ lives in the union Ω_k of k-dimensional subspaces, the union Ω_k thus includes C_n^k subspaces. To stably recover the k-sparse signal x from m measurements, the measurement matrix Φ is required to satisfy the Restricted Isometry Property (RIP) [3].

Definition:(k-RIP) *A matrix $\Phi \in \mathbb{R}^{m\times n}$ is said to have k-restricted isometry property (k-RIP) with constant $\delta_k > 0$ if, for all x in the union Ω_k,*

$$(1-\delta_k) \parallel x \parallel_2^2 \leq \parallel \Phi x \parallel_2^2 \leq (1+\delta_k) \parallel x \parallel_2^2 \qquad (3)$$

While the sparse signal x lives in a union of subspaces $\mathcal{A} \subset \mathbb{R}^n$, the k-RIP can be extended to the $\mathcal{A}$-RIP [2]:

Definition:($\mathcal{A}$-RIP) *A matrix $\Phi \in \mathbb{R}^{m\times n}$ is said to have $\mathcal{A}$-restricted isometry property ($\mathcal{A}$-RIP)with constant $\delta_{\mathcal{A}}(\Phi)$ if, for all x living in the union of subspaces $\mathcal{A}$*

$$(1-\delta_{\mathcal{A}}(\Phi)) \parallel x \parallel_2^2 \leq \parallel \Phi x \parallel_2^2 \leq (1+\delta_{\mathcal{A}}(\Phi)) \parallel x \parallel_2^2 \qquad (4)$$

Blumensath and Davies have proposed one theorem on the sufficient condition for stable sparse recovery to guide the minimal measurement number m necessary for a subgaussian random measurement matrix to have the $\mathcal{A}$-RIP with the given probability [2]:

Lemma 1: *Suppose $\mathcal{A}_k \subset \mathbb{R}^n$ be the union of L subspaces of k-dimensions aligned with $\mathbb{R}^n$. For any $t > 0$, if*

$$m \geq \frac{2}{c\delta_{\mathcal{A}_k}}(log(2L) + klog(\frac{12}{\delta_{\mathcal{A}_k}}) + t) \qquad (5)$$

then the subgaussian random matrix $\Phi \in \mathbb{R}^{m\times n}$ has the $\mathcal{A}$-RIP with constant $\delta_{\mathcal{A}_k}$, where $0 < \delta_{\mathcal{A}_k} < 1$, and $c > 0$ only depends on the $\delta_{\mathcal{A}_k}$. The probability is at least $1-e^{-t}$.

For the k-sparse data recovery, we know that $\mathcal{A}_k \subset \mathbb{R}^n$ is the union of $L = C_n^k$ subspaces. Thus, this theorem directly leads to the classic CS result $m = \mathcal{O}(k + klog(n/k))$.

3. Dynamic Group Sparsity

The success of sparse recovery in compressive sensing motivates us to further observe the support set of the sparse coefficients. Observations on some practical sparse data show that the support sets often have the group clustering trend with dynamic and unknown group structure. Intuitively, the measurement number bound may be further reduced if this trend can be dexterously utilized as sparsity in convention CS. In this section, we propose a new algorithm to seamlessly combine this prior with sparsity, which is shown to enable better recovery results for this case with less measurement requirement and lower computation complexity.

3.1. Dynamic Group Sparse Data

Similar to the definition of k-sparse data, we can define dynamic group sparse data as follow:

Definition:($G_{k,q}$-sparse data) *A data $x \in \mathbb{R}^n$ is defined as the dynamic group sparse data ($G_{k,q}$-sparse data) if it can be well approximated using $k \ll n$ nonzero coefficients under some linear transforms and these k nonzero coefficients are clustered into $q \in \{1, \cdots, k\}$ groups.*

From this definition, we can know that $G_{k,q}$-sparse data only requires that the nonzero coefficients in the sparse data have the group clustering trend and does not require to know any information about the group size and location. In the following, it will be further illustrated that the group number q is also not necessary to be known in our algorithm. The group structures can be dynamic and unknown. Figure 1 shows a real sample of $G_{k,q}$-sparse data in a video surveillance application. We can find that nonzero coefficients are not randomly distributed but clustered spatially in the background subtracted image (Figure 1 (b)) and the foreground mask (Figure 1 (c)). More specially, the R, G and B channels of the background subtracted image share a common support set although the nonzero coefficients are spatially clustered in each channel respectively.

(a) (b) (c) (d)

Figure 1. An example of dynamic group sparse data: (a) one video frame, (b) the foreground image, (c) the foreground mask and (d) the background subtracted image with the proposed algorithm.

Due to the additional dynamic clustering prior, the union of subspaces containing $G_{k,q}$-sparse data does not span all k-dimensional subspaces of the union Ω_k as in the conventional CS [4, 8, 6, 18, 23]. The former is far narrower than the latter in most cases. The dynamic group clustering prior significantly reduces the degrees of freedom of the sparse signal since it only permits certain combinations of its support set rather than all random combinations. This will make it possible for us to decrease the minimal measurement number m for stable recovery. Therefore, we need to develop a new theorem on the sufficient condition for stable dynamic group sparse data recovery.

Lemma 2: *Suppose $G_{k,q}$-sparse data $x \in \mathbb{R}^n$ is known to live in $L_{k,q}$ k-dimensional subspaces aligned to $\mathbb{R}^n$, and $\mathcal{G}_{k,q}$ is the smallest union of these k-dimension subspaces. For any $t > 0$, if*

$$m = \mathcal{O}(k + qlog(n/q)) \tag{6}$$

then the subgaussian random matrix $\Phi \in \mathbb{R}^{m \times n}$ has the $\mathcal{A}$-RIP with constant $\delta_{\mathcal{A}_k}$, where $0 < \delta_{\mathcal{A}_k} < 1$. The probability is at least $1 - e^{-t}$.

Proof: According to Lemma 1, the key point of the proof is to find the value or upper bound of $L_{k,q}$. After we obtain its value or upper bound, we can directly obtain the upper bound of the measurement number m according to equation 5 in Lemma 1. Suppose the dimension of the sparse signal is d. We know that k nonzero coefficients of $G_{k,q}$-sparse data $x \in \mathbb{R}^n$ have clustered into q groups. Suppose that the i-th group has r_i nonzero coefficients. Thus, $r_i, i = 1, \cdots, q$ should be a natural number and their sum is k. Without loss of generality, we can assume that n coefficients of $x \in \mathbb{R}^n$ are divided into q regions, where the i-th region has $\frac{nr_i}{k}$ nonzero coefficients and every region is not overlapped with other regions (While n/k is not an integer, we can round them off and just keep their sum with n). Considering the restrictions that $r_i, i = 1, \cdots, q$ should be a natural number and their sum is k, there are C_{k-1}^{q-1} possible combinations. According to Stirling's formula, $C_{k-1}^{q-1} \leq e^q (k/q)^{q-1}$. For each combination, we divide the original problem into q small problems. For each small problem, it is equal to the case where all r_i nonzero coefficients are clustered into one group. Thus, $L_{r_i,1} \leq \frac{nr_i}{k}(2d-1)^{r_i - 1}$ for each small problem. Then, the number of all possible combinations for the original problem is:

$$\begin{aligned} L_{k,q} &\leq (\prod_{i=1}^{q} (\frac{nr_i}{k})(2d-1)^{r_i - 1}) C_{k-1}^{q-1} \\ &\leq (\frac{n}{q})^q (2d-1)^{k-q} e^q (k/q)^{q-1} \end{aligned} \tag{7}$$

If we do consider the overlapping problems between the nonzero coefficients of neighboring regions, the number of all allowed combinations for d-dimensional $G_{k,q}$-sparse data should be further less than the right of this equation. But the bound in equation 7 is enough for our proof. According to equation 5 in Lemma 1, we know:

$$m \geq \frac{2}{c\delta_{\mathcal{A}_k}} (log(2L_{k,q}) + klog(\frac{12}{\delta_{\mathcal{A}_k}}) + t) \tag{8}$$

Thus, we can easily obtain $m = \mathcal{O}(k + qlog(n/q))$, which proves the Lemma.

Lemma 2 shows that the number of measurements required for robustly recovering dynamic group sparsity data is $m = \mathcal{O}(k + qlog(n/q))$, which is a significant improvement over the $m = \mathcal{O}(k + klog(n/k))$ that would be required by conventional CS recovery algorithms [4, 8, 18, 23]. While the group number q is smaller, more improvements can be obtained. While q is far smaller than k and k is close to $log(n)$, we can get $m = \mathcal{O}(k)$. Note that, this is a sufficient condition. If we know more priors about group settings, we can further reduce this bound.

3.2. Dynamic Group Sparsity Recovery

Lemma 2 equips us to propose a new recovery algorithm for dynamic group sparse data, namely dynamic group sparsity (DGS) recovery algorithm. From the introduction, we know that only the SP [6] and the CoSaMP [18] have better balance between theoretical guarantee and computation complexity among existing greedy recovery algorithms. Actually, these two algorithms have a similar framework. In this section, we demonstrate how to seamlessly integrate the dynamic group clustering prior into that framework.

Our algorithm includes five main steps in each iteration: 1) pruning the residue estimation; 2) merging the support sets; 3) estimating the signal by least square; 4) pruning the signal estimation and 5) updating the signal/residue estimation and support set. One can observe that it is similar to that of SP/CoSaMP algorithms. The difference only exists in the pruning process in step 1 and step 4. The modification is simple. We prune the estimation in the step 1 and step 4 using DGS approximation pruning rather than k-sparse approximation, as we only need to search over subspaces of $\mathcal{A}_{k,q}$ instead of C_n^k subspaces of Ω_k. It directly leads to fewer measurement requirement for stable data recovery.

The DGS pruning algorithm is described in algorithm 1. There exist two prior-dependent parameters J_y and J_b. J_y is the number of tasks if the problem can be represented as a multi-task CS problem [13]. J_b is the block size if the interested problem can be modelled as a block sparsity problem [1, 24, 22, 26]. Their default values are set as 1, which is the case of traditional sparse recovery in compressive sensing. Moreover, there are two important user-tuning parameters, the weight w of neighbors and the neighbor number τ of each element in sparse data. In practice, it is very straightforward to adjust them since they have the physical meanings. The first one controls the balance between the sparsity prior and the group clustering prior. While w is smaller/bigger, it means that the degree of dynamic group clustering is lower/higher in the sparse signal. Generally, they are set as $0.5's$ if there are not more knowledge about that in practice. The parameter τ controls the number of neighbors that can be affected by each element in sparse data. Generally, it is good enough to set it as 2, 4 and 6 for 1D, 2D and 3D data respectively.

Algorithm 1. ***DGS approximation pruning***

Input: $x \in \mathbb{R}^n$ {estimations}; k {the sparsity number}; J_y {task number}; J_b {block size}; $N_x \in \mathbb{R}^{n\times\tau}$ {values of x's neighbors}; $w \in \mathbb{R}^{n\times\tau}$ {weights for neighbors}; τ {neighbor number}

$J_x = J_y J_b$;$x \in \mathbb{R}^n$ is shaped to $x \in \mathbb{R}^{\frac{n}{J_x}\times J_x}$
$N_x \in \mathbb{R}^{n\times\tau}$ is shape to $N_x \in \mathbb{R}^{\frac{n}{J_x}\times J_x\times\tau}$;
for all $i = 1, ..., \frac{n}{J_x}$ **do**

Combing each entry with its neighbors

$$z(i) = \sum_{j=1}^{J_x} x^2(i,j) + \sum_{j=1}^{J_x}\sum_{t=1}^{\tau} w^2(i,t)N_x^2(i,j,t)$$

end for
$\Omega \in \mathbb{R}^{\frac{n}{J_x}\times 1}$ is set as indices corresponding to the largest k/J_x entries of z
for all $j = 1, ..., J_x$ **do**
  for all $i = 1, ..., \frac{k}{J_x}$ **do**

Obtain the final list
$\Gamma((j-1)\frac{k}{J_x} + i) = (j-1)\frac{k}{J_x} + \Omega(i)$
  end for
end for
Output: $supp(x, k) \leftarrow \Gamma$

Up to now, we assume that we know the sparsity number k of the sparse data before recovery. However, it is not always true in practical applications. For example, we do not know the exact sparsity numbers of the background subtracted images although we know they tend to be dynamic group sparse. Motivated by the idea in [7], we develop a new recovery algorithm called AdaDGS by incorporating an adaptive sparsity scheme into the above DGS recovery algorithm.

Suppose the range of the sparsity number is known to be $[k_{min}, k_{max}]$. We can set the step size of sparsity number as $\triangle k$. The whole recovery process is divided into several stages, each of which includes several iterations. Thus, there are two loops in AdaDGS recovery algorithm. The sparsity number is initialized as k_{min} before iterations. During each stage (inner loop), we iteratively optimize sparse data with the fixed sparsity number k_{curr} until the halting condition within the stage is true (for example, the residue norm is not decreasing). We then switch to the next stage after adding $\triangle k$ into the current sparsity number k_{curr} (outer loop). The whole iterative process will stop whenever the halting condition is satisfied. For practical applications, there is a trade-off between the sparsity step size $\triangle k$ and the recovery performance. Smaller step sizes require more iterations and bigger step size may cause in-

Algorithm 2. ***AdaDGS Recovery***

Input: $\Phi \in \mathbb{R}^{m\times n}$ {sample matrix}; $y \in \mathbb{R}^m$ {sample vector}; $[k_{min}, k_{max}]$ {sparsity range}; $\triangle k$ {sparsity step size}

Initialization: residue $y_r = y$; $\Gamma = supp(x) = \varnothing$; sparse data $x = 0$; sparsity number $k = k_{min}$

repeat

  Perform DGS recovery algorithm with sparsity number k to obtain x and the residue

  if halting criterion false **then**

    Update Γ, y_r and $k = k + \triangle k$

  end if

until halting criterion true

Output: $x = \Phi_\Gamma^\dagger y$

accuracy. The sparsity range depends on the applications. Generally, it can be set as $[1, n/3]$, where n is the dimension of the sparse data. Algorithm 2 describes the proposed AdaDGS recovery algorithm.

3.3. AdaDGS Background Subtraction

Background subtraction is an important pre-processing step in video monitoring applications. There exist a lot of methods for this problem. The Mixture of Gaussians (MoG) background model assumes the color evolution of each pixel can be modelled as a MoG and are widely used on realistic scenes [21]. Elgammal et al. [10] proposed a non-parametric model for the background under similar computational constraints as the MoG. Spatial constraints are also incorporated into their model. Sheikh and Shah consider both temporal and spatial constraints in a Bayesian framework [20], which results in good foreground segmentations even when the background is dynamic. The model in [16] also uses a similar scheme. All these methods only implicitly model the background dynamics. In order to better handle dynamic scenes, some recent works [17, 27] explicitly model the background as dynamic textures. Most dynamic texture modeling methods are based on the Auto Regressive and Moving Average (ARMA) model, whose dynamics is driven by a linear dynamic system (LDS). While this linear model can handle background dynamics with certain stationarity, it will cause over-fitting for more complex scenes.

The inspiration for our AdaDGS background subtraction came from the success in online DT video registration based on the sparse representation constancy assumption (SRCA) [12]. The SRCA states that a new coming video frame should be represented as a linear combination of as few preceding image frames as possible. As a matter of fact, the traditional brightness constancy assumption seeks that the current video frame can be best represented by a single preceding frame, while the SRCA seeks that the current frame can be best sparsely represented by all preceding image frames. Thus, the former can be thought as a special case of SRCA.

Suppose a video sequence consists of frames $I_1, ..., I_n \in \mathbb{R}^m$. Without loss of generality, we can assume that background subtraction has already been performed on the first t frames. Let $A = [I_1, ..., I_t] \in \mathbb{R}^{m\times t}$. Denote the background image and the background subtracted image by b and f, respectively, for I_{t+1}. From the introduction in Section 3.1, we know that f is dynamic group sparse data with unknown sparsity number k_f and group structure. According to SRCA, we have $b = Ax$, where $x \in \mathbb{R}^t$ should be k_x-sparse vector and $k_x << t$. Let $\Phi = [A, I] \in \mathbb{R}^{m\times(t+m)}$, where $I \in \mathbb{R}^{m\times m}$ is an identity matrix. Then, we have:

$$I_{t+1} = Ax + f = [A, I] \begin{bmatrix} x \\ f \end{bmatrix} = \Phi z \tag{9}$$

where $z \in \mathbb{R}^{t+m}$ is the DGS data with unknown sparsity $k_x + k_f$. Background subtraction is thus formulated as the following AdaDGS recovery problem:

$$(x_0, f_0) = argmin\|z\|_0, \quad \|I_{t+1} - \Phi z\|^2 < \varepsilon \tag{10}$$

which can be efficiently solved by the proposed AdaDGS recovery algorithm. Similar ideas are used for face recognition robust to occlusion [25]. It is worth mentioning that the coefficients in w corresponding to the x part are randomly sparse while those corresponding to f are dynamic group sparse. During the DGS approximation pruning, we thus can set those coefficients in weight w for the x-related part as zeros and those for f as nonzeros. Since we do not know the sparsity number k_x and k_f, we can set sparsity ranges for them respectively and run the AdaDGS recovery algorithm until the halting condition is true. Then, we can obtain the optimized background subtracted image f and background image $b = Ax$. For long video sequences, it is impractical to build a model matrix $A = [I_1, ..., I_t] \in \mathbb{R}^{m\times t}$, where t denotes the last frame number. In order to cope with this case, we can set a time window width parameter τ. We then build the model matrix, $A = [I_{t-\tau+1}, ..., I_t] \in \mathbb{R}^{m\times(t-\tau)}$, for the $(t+1)$ frame, which can avoid the memory requirement blast for a long video sequence. The complete algorithm for AdaDGS based background subtraction is summarized in Algorithm 3.

4. Experiments

For quantitative evaluation, the recovery error is defined to indicate the difference between the estimation x_{est} and the ground-truth x:$\|x_{est} - x\|_2/\|x\|_2$. All experiments are conducted on a $3.2GHz$ PC in Matlab environment.

Algorithm 3. ***AdaDGS Background Subtraction***

Input: The video sequence $I_1, ..., I_n$, the number t which means $1^{st} \sim t^{th}$ have been performed background subtraction, the time window width $\tau \leq t$
for all $j = t+1, ..., n$ **do**
  Set $A = [I_{j-\tau}, ..., I_{j-1}]$ and form $\Phi = [A, I]$
  Set $y = I_j$ and the sparsity ranges/step-sizes
  $(x_0, f_0) = AdaDGS(\Phi, y)$
end for
Output: Background subtracted images

4.1. 1D Simulated Signals

In the first experiment, we randomly generate a $1D$ $G(k,q)$-sparse signal with values ± 1, where $n = 512$, $k = 64$ and $q = 4$. The projection matrix Φ is generated by creating a $m \times n$ matrix with i.i.d. draws of a Gaussian distribution $N(0;1)$, and then the rows of Φ are normalized to the unit magnitude. Zero-mean Gaussian noise with standard deviation $\sigma = 0.01$ is added to the measurements. Figure 2 shows one generated signal and its recovered results by different algorithms when $m = 3k = 192$. As the measurement number m is only 3 times of the sparsity number k, both of other algorithms can not obtain good recovery results, whereas the DGS obtains almost perfect recovery results with the least running time. To study how the measurement number m effects the recovery performance, we change the measurement number and record the recovery results by different algorithms. To reduce the randomness, we execute the experiment 100 times for each of the measurement numbers in testing each algorithm. Figure 3 shows the performance of 5 algorithms with increasing measurements in terms of the recovery error and running time. Overall, the DGS obtains the best recovery performance with the least computation; the recovery performance of GPSR, SPGL1-Lasso and SP is close; and the l^1-norm minimization based GPSR and SPGL1-Lasso require more computation than greedy algorithms such as the OMP, SP and our DGS. In the three greedy algorithms, the OMP has the worst recovery performance. All these experimental results are consistent with our theorem: the proposed DGS algorithm can achieve better recovery performance for DGS data with far few measurements and less computation complexity.

4.2. 2D Color Images

To validate the proposed recovery algorithm on 2D images, we randomly generate a $2D$ $G(k,q)$-sparse color image by putting four digits in random locations, where $n = H * W = 48 * 48$, $k = 152$ and $q = 4$. The projection matrix Φ and noises are generated with the similar method as that for 1D signal. The $G(k,q)$-sparse color image has a special property: the R, G and B channels share a common support set, while the nonzero coefficients have dynamic group clustering trends within each channel. Thus, the recovery of DGS color images can be considered as a 3-task CS recovery problem. As for the input parameters of DGS in this case, we just need to set J_y as 3 and keep other default parameters unchanged. Considering the MCS is specially designed for multi-task CS problems, we will compare it with DGS and SP. Figure 4 shows one example 2D $G(k,q)$-sparse color image and the recovered results by different algorithms when $m = 440$. Figure 5 shows the performance of the three algorithm, averaged over 100 random runs for each sample size. The DGS achieves the best recovery performance with far less computation. It is easily understood because DGS exploits three priors for recovery: (1) the three color channels share a common support set, (2) there are dynamic group clustering trends within each color channel and (3) sparsity prior exists in each channel; thus it achieves better results than MCS, which only uses two priors. The SP is the worst since it only uses one prior. This experiment clearly demonstrates: the more valid priors are used for sparse recovery, the more accurate results we can achieve. That is the main reason why DGS, MCS and SP obtained the best, good and the worst recovery results.

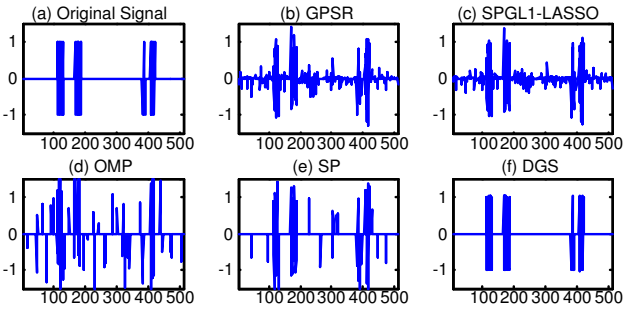
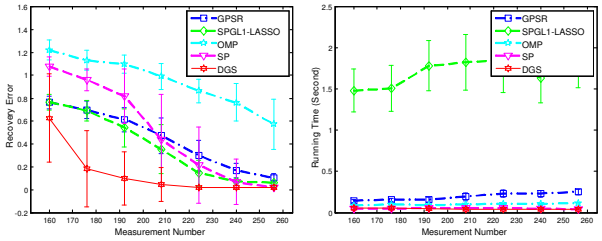


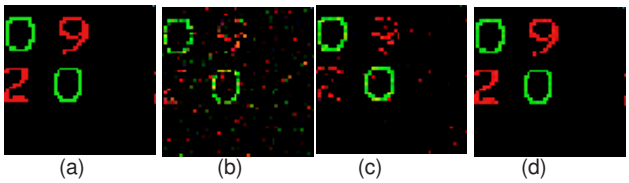
Figure 2. Recovery results of 1D data. (a) Original data; (b) GPSR (error is 0.5173 and time is 0.1847 seconds); (c) SPGL1-Lasso (error is 0.4021 and time is 1.1497 seconds); (d) OMP (error is 1.0270 and time is 0.1422 seconds);(e) SP (error is 0.6143 and time is 0.1100 seconds);(f) DGS recovery (error is 0.0178 and time is 0.0605 seconds).



(a) (b)

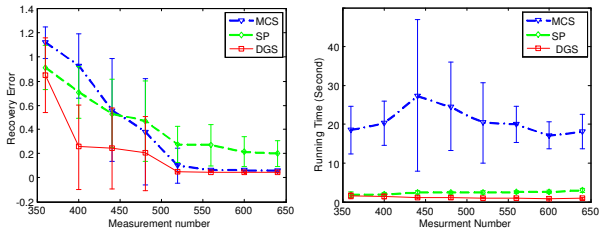
Figure 3. Recovery errors vs. measurement numbers: a) recovery errors; (b) running times

Figure 5 (b) shows the comparison of running times by the three algorithms. It is not surprising that the running times with DGS are always far less than those with MCS and a little less than those with SP for all measurement numbers.



(a) (b) (c) (d)

Figure 4. Recovery results of a 2D color image: (a) original color image, (b) recovered image with MCS [13] (error is 0.8399 and time is 29.2656 seconds), (c) recovered image with SP [6] (error is 0.7605 and time is 1.6579 seconds) and (d) recovered image with DGS (error is 0.1176 and time is 1.0659 seconds).



(a) (b)

Figure 5. Recovery performance: a) errors; (b) running times

4.3. Background Subtraction

Background subtracted images are typical $G(k, q)$-sparse data. They generally correspond to the interested foreground objects. Compared with the whole scene, they tend to be not only spatially sparse but also cluster into dynamic groups, although the sparsity number and group structures are not known. As we know, the sparsity number must be provided in most of current recovery algorithms, which make them impractical for this problem. In contrast, the proposed AdaDGS can apply well to this task since it not only can automatically learn the sparsity number and group structures but also is a fast enough greedy algorithm.

The first experiment is designed to validate the advantage of the AdaDGS model. We test the proposed algorithm on Zhong's dataset [27]. The background subtracted images can be directly obtained with the proposed AdaDGS. The corresponding binary mask of these images are obtained with the simple threshold. The Zhong'results with robust Kalman model are also shown for comparisons. Figure 6 shows the results. Note that all results with AdaDGS are not post-processed with morphological operations and the results are directly the solutions of the optimization problem in Equation 10. It is clear that our AdaDGS produces clean background subtracted images, which shows the advantages of the DGS model. Figure 7 and Figure 8 show the background subtraction results on two other videos [10, 17]. Note that our results without postprocessing can compete with others with postprocessing. The results show the proposed AdaDGS model can handle well highly dynamic scenes by exploiting the effective sparsity optimization scheme.

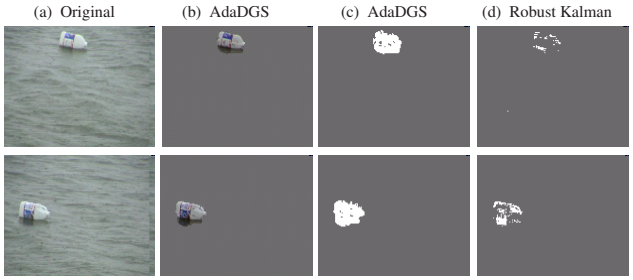


Figure 6. Results on the Zhong's dataset (a) original frame, (b) background subtracted image with AdaDGS, (c) the binary mask with AdaDGS and d) with robust Kalman model [27]

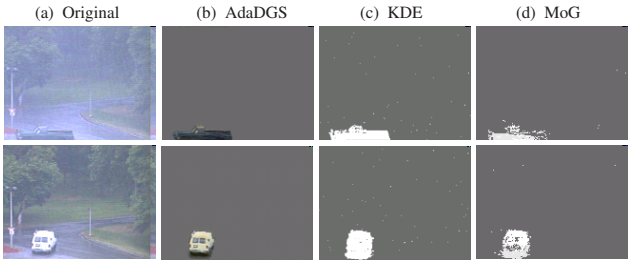


Figure 7. Results on the Elgammal's dataset. (a) original frame, (b) with AdaDGS, (c) with KDE model [10] (d) with MoG [21]

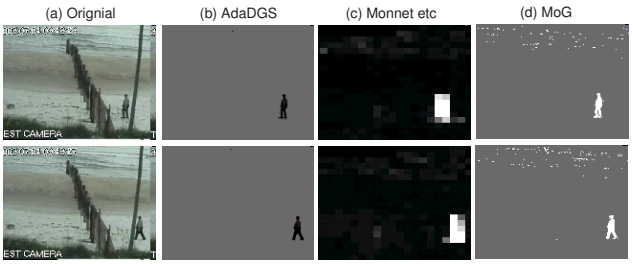


Figure 8. Results on Monnet's dataset. (a) original frame, (b) with AdaDGS, (c) with Monnet's method [17] and (d) with MoG [21]

4.4. Discussion

All experimental results show the proposed algorithms gain marked improvement over previous algorithms when DGS priors are available. From a practical perspective, the proposed DGS/AdaDGS can recover DGS data with higher accuracy and lower computational complexity from fewer measurements. From a theoretical point of view, Lemma 2 offers a stronger guarantee for DGS/AdaDGS to achieve

stable recovery. Moreover, we provide a generalized framework for priors-driven sparse data recovery algorithms. Using different input parameter settings, it can perform sparse recovery, multi-task sparse recovery, group/block sparse recovery, DGS recovery, and adaptive DGS recovery, respectively. Group structure and sparsity number are not must-knows for our algorithm, which makes it flexible and applicable in many practical applications; as far as we know, this property of our algorithm is unique among all existing sparse recovery algorithms.

5. Conclusions

In this paper, we extend the theory of CS to efficiently handle dynamic group sparse data. Based on this extended theory, the proposed algorithm can stably recover dynamic group-sparse data using far fewer measurements and less computation than the current state-of-the-art algorithms with provable guarantees. It has been applied to sparse recovery and background subtraction on both simulated and practical data. Experimental results demonstrate the performance guarantee of the proposed algorithm and show marked improvement over previous algorithms.

References

[1] E. Berg, M. Schmidt, M. Friedlander, and K. Murphy. Group sparsity via linear-time projection. 2008. Preprint. 2, 4

[2] T. Blumensath and M. Davies. Sampling theorems for signals from the union of finite-dimensional linear subspaces. *IEEE Transactions on Information Theory*, 2008. Accepted. 2

[3] E. Candes. Compressive sampling. In *Proceedings of the International Congress of Mathematicians*, 2006. 2

[4] E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52:489–509, 2006. 1, 3, 4

[5] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scienticifc Computing*, 61:20–33, 1998. 1

[6] W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing: closing the gap between performance and complexity, 2008. Preprint. 1, 3, 4, 7

[7] T. Do, L. Gan, N. Nguyen, and T. Tran. Sparsity adaptive matching pursuit algorithm for practical compressed sensing. 2008. Accepted. 4

[8] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52:1289–1306, 2006. 1, 2, 3, 4

[9] D. Donoho, Y. Tsaig, I. Drori, and J. Starck. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit, 2007. Submitted. 1

[10] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *Proceedings of ECCV*, 2000. 5, 7

[11] M. Figueiredo, R. Nowak, and S. Wright. Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. *IEEE Journal on Selected Topics in Signal Processing*, 1(4):586–597, 2007. 1

[12] J. Huang, X. Huang, and D. Metaxas. Simultaneous image transformation and sparse representation recovery. In *Proceedings of CVPR*, 2008. 5

[13] S. Ji, D. Dunson, and L. Carin. Multi-task compressive sensing. *IEEE Transactions on Signal Processing*, 2008. Submitted. 2, 4, 7

[14] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. A method for large-scale l1-regularized least squares. *IEEE Journal on Selected Topics in Signal Processing*, 1(4):606–607, 2007. 1

[15] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. 1

[16] A. Mittal and M. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *Proceedings of CVPR*, 2004. 5

[17] A. Monnet, A. Mittal, N. Paragios, and Y. Ramesh. Background modeling and subtraction of dynamic scenes. In *Proceedings of ICCV*, 2003. 5, 7

[18] D. Needell and J. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 2008. Accepted. 1, 3, 4

[19] D. Needell and R. Vershynin. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit, 2007. Submitted. 1

[20] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 27, 2005. 5

[21] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. 2:246–252, 1999. 5, 7

[22] M. Stojnic, F. Parvaresh, and B. Hassibi. On the reconstruction of block-sparse signals with an optimal number of measurements. 2008. Preprint. 2, 4

[23] J. Tropp and A. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007. 1, 3, 4

[24] J. Tropp, A. Gilbert, and M. Strauss. Algorithms for simultaneous sparse approximation. part i: Greedy pursuit. *IEEE Transactions on Signal Processing*, 86:572–588, 2006. 2, 4

[25] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009. 5

[26] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of The Royal Statistical Society Series B*, 68(1):49–67, 2006. 2, 4

[27] J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In *Proceedings of ICCV*, 2003. 5, 7