Fast Optimization for Mixture Prior Models

Junzhou Huang, Shaoting Zhang, and Dimitris Metaxas

Division of Computer and Information Sciences, Rutgers University, USA
{jzhuang, shaoting, dnm}@cs.rutgers.edu

Abstract. We consider the minimization of a smooth convex function regularized by the mixture of prior models. This problem is generally difficult to solve even each simpler regularization problem is easy. In this paper, we present two algorithms to effectively solve it. First, the original problem is decomposed into multiple simpler subproblems. Then, these subproblems are efficiently solved by existing techniques in parallel. Finally, the result of the original problem is obtained from the weighted average of solutions of subproblems in an iterative framework. We successfully applied the proposed algorithms to compressed MR image reconstruction and low-rank tensor completion. Numerous experiments demonstrate the superior performance of the proposed algorithm in terms of both the accuracy and computational complexity.

Key words: Compressive Sensing, MRI Reconstruction, Tensor Completion

1 Introduction

The mixture of prior models have been used in many fields including sparse learning, computer vision and compressive sensing. For example, in compressive sensing, the linear combination of the total-variation (TV) norm and L1 norm is known as the most powerful regularizer for compressive MR imaging [1–3] and widely used in recovering the MR images.

In this paper, we propose two composite splitting algorithms to solve this problem:

$$min\{F(x) \equiv f(x) + \sum_{i=1}^{m} g_i(B_i x), x \in \mathbf{R^p}\} \tag{1}$$

where f is the loss function and $\{g_i\}_{i=1,...,m}$ are the prior models; f and $\{g_i\}_{i=1,...,m}$ are convex functions and $\{B_i\}_{i=1,...,m}$ are orthogonal matrices. If the functions f and $\{g_i\}_{i=1,...,m}$ are well-structured, there are two classes of splitting algorithms to solve this problem: operator splitting and variable splitting algorithms.

The operator-splitting algorithm is to search an x to make the sum of the corresponding maximal-monotone operators equal to zero. Forward-Backward schemes are widely used in operator-splitting algorithms [4–6]. These algorithms have been applied in sparse learning [7] and compressive MR imaging [2]. The Iterative Shrinkage-Thresholding Algorithm (ISTA) and Fast ISTA (FISTA) [8] are two important Forward-Backward methods. They have been successfully used in signal processing [8, 9], matrix completion [10] and multi-task learning [11].

The variable splitting algorithm is another choice to solve problem (1) based on the combination of alternating direction methods (ADM) under an augmented Lagrangian framework. It was firstly used to solve the numerical PDE problem in [12, 13]. Tseng and He et al. extended it to solve variational inequality problems [14, 15]. There has been a lot of interests from the field of compressive sensing [16, 17], where L1 regularization is a key problem and can be efficiently solved by this type of algorithms [18–20]. It also shows the effectiveness for the sparse covariance selection problem in [21]. The Multiple Splitting Algorithm (MSA) and Fast MSA (FaMSA) have been recently proposed to efficiently solve (1), while $\{g_i\}_{i=1,\ldots,m}$ are assumed to be smooth convex functions [22].

However, all these algorithms can not efficiently solve (1) with provable convergence complexity. Moreover, none of them can provide the iteration complexity bounds for their problems, except ISTA/FISTA in [8] and MSA/FaMSA in [22]. Both ISTA and MSA are first order methods. Their complexity bounds are $O(1/\epsilon)$ for ϵ-optimal solutions. Their fast versions, FISTA and FaMSA, have complexity bounds $O(1/\sqrt{\epsilon})$ correspondingly, which are inspired by the seminal results of Nesterov and are optimal according to the conclusions of Nesterov [23, 24]. However, Both ISTA and FISTA are designed for simpler regularization problems and can not be applied efficiently to the composite regularization problem in our formulation. While the MSA/FaMSA in [22] are designed to handle the case of $m \geq 1$ in (1), they assume that all $\{g_i\}_{i=1,\ldots,m}$ are smooth convex functions, which make them unable to directly solve the problem (1). Before applying them, we have to smooth the nonsmooth function $\{g_i\}_{i=1,\ldots,m}$ first. Since the smooth parameters are related to ϵ, the FaMSA with complexity bound $O(1/\sqrt{\epsilon})$ requires $O(1/\epsilon)$ iterations to compute an ϵ-optimal solution, which means that it is not optimal for this problem.

In this paper, we propose two splitting algorithms based on the combination of variable and operator splitting techniques. We dexterously decompose the hard composite regularization problem (1) into m simpler regularization subproblems by: 1) splitting the function $f(x)$ into m functions $f_i(x)$ (for example: $f_i(x) = f(x)/m$); 2) splitting variable x into m variables $\{x_i\}_{i=1,\ldots,m}$; 3) performing operator splitting to minimize $h_i(x_i) = f_i(x_i) + g_i(B_i x_i)$ over $\{x_i\}_{i=1,\ldots,m}$ independently and 4) obtaining the solution x by the linear combination of $\{x_i\}_{i=1,\ldots,m}$. This includes both function splitting, variable splitting and operator splitting. We call them Composite Splitting Algorithms (CSA) and fast CSA (FCSA). Compared to ISTA and MSA, CSA is more general as it can efficiently solve composite regularization problems with m ($m \geq 1$) nonsmooth functions. More importantly, our algorithms can effectively decompose the original hard problem into multiple simpler subproblems and efficiently solve them in parallel. Thus, the required CPU time is not longer than the time required to solve the most difficult subproblem using current parallel-processor techniques.

The remainder of the paper is organized as follows. Section 2 briefly reviews the related algorithms. The composite splitting algorithm and its accelerated version are proposed to solve problem (1) in section 3. Numerical experiment results are presented in Section 4. Finally, we provide our conclusions in Section 5.

Algorithm 1 ISTA

```
Input: $\rho = 1/L_f, x_0$
repeat
  for $k = 1$ to $K$ do
    $x^k = prox_\rho(g)(x^{k-1} - \rho \nabla f(x^{k-1}))$
  end for
until Stop criterions
```

2 Algorithm Review

2.1 Notations

We provide a brief summary of the notations used throughout this paper.

Matrix Norm and Trace:

1. Operator norm or 2-norm: $||X||$;
2. L_1 and Total Variation norm: $||X||_1$ and $||X||_{TV}$;
3. Matrix inner product: $\langle X, Y \rangle = trace(X^H Y)$.

Gradient: $\nabla f(x)$ denotes the gradient of the function f at the point x.

The proximal map: given a continuous convex function $g(x)$ and any scalar $\rho > 0$, the proximal map associated to function g is defined as follows [9, 8]:

$$prox_\rho(g)(x) := \arg\min_u \{g(u) + \frac{1}{2\rho}\|u - x\|^2\} \tag{2}$$

ϵ-optimal Solution: Suppose x^* is an optimal solution to (1). $x \in \mathbf{R}^p$ is called an ϵ-optimal solution to (1) if $F(x) - F(x^*) \le \epsilon$ holds.

2.2 ISTA and FISTA

The ISTA and FISTA consider the following optimization problem [8]:

$$min\{F(x) \equiv f(x) + g(x), x \in \mathbf{R^p}\} \tag{3}$$

Here, they make the following assumptions:

1. $g : \mathbf{R^p} \rightarrow \mathbf{R}$ is a continuous convex function, which is possibly nonsmooth;
2. $f : \mathbf{R^p} \rightarrow \mathbf{R}$ is a smooth convex function of type $C^{1,1}$ and the continuously differential function with Lipschitz constant L_f: $\|\nabla f(x_1) - \nabla f(x_2)\| \le L_f \|x_1 - x_2\|$ for every $x_1, x_2 \in \mathbf{R^p}$;
3. Problem (3) is solvable.

Algorithm 1 outlines the ISTA. Beck and Teboulle show that it terminates in $O(1/\epsilon)$ iterations with an ϵ-optimal solution in this case.

Theorem 1. *(Theorem 3.1 in [8]): Suppose $\{x_k\}$ is iteratively obtained by the algorithm of the ISTA, then, we have*

$$F(x^k) - F(x^*) \le \frac{L_f \|x^0 - x^*\|^2}{2k}, \forall x^* \in X_*$$

Algorithm 2 FISTA

```
Input: $\rho = 1/L_f$, $r^1 = x^0$, $t^1 = 1$
repeat
    for $k = 1$ to $K$ do
        $x^k = prox_\rho(g)(r^k - \rho\nabla f(r^k))$
        $t^{k+1} = \frac{1+\sqrt{1+4(t^k)^2}}{2}$
        $r^{k+1} = x^k + \frac{t^k-1}{t^{k+1}}(x^k - x^{k-1})$
    end for
until Stop criterions
```

Algorithm 2 outlines the FISTA. Compared with ISTA, the increased computation burdens come from the second step and third step in each iteration, which is almost negligible in large scale applications. Because of these advantages, the key idea of the FISTA is recently widely used in large scale applications, such as compressive sensing [8], image denoising and deblurring [9], matrix completion [10] and multi-task learning [11]. It has been proven that (Theorem 4.1 in [8]), with this acceleration scheme, the algorithm can terminate in $O(1/\sqrt{\epsilon})$ iterations with an ϵ-optimal solution instead of $O(1/\epsilon)$ for those of ISTA.

Theorem 2. *(Theorem 4.1 in [8]): Suppose $\{x^k\}$ and $\{r^k\}$ are iteratively obtained by the FISTA, then, we have*

$$F(x^k) - F(x^*) \leq \frac{2L_f\|x^0 - x^*\|^2}{(k+1)^2}, \forall x^* \in X_*$$

The efficiency of the FISTA highly depends on being able to quickly solve their first step $x^k = prox_\rho(g)(x_g)$, where $x_g = r^k - \rho\nabla f(r^k)$. For simpler regularization problems, it is possible, i.e, the FISTA can rapidly solve the $L1$ regularization problem with cost $\mathcal{O}(p\log(p))$ [8] (where n is the dimension of x), since the second step $x^k = prox_\rho(\beta\|\Phi x\|_1)(x_g)$ has a close form solution; It can also quickly solve the TV regularization problem, since the step $x^k = prox_\rho(\alpha\|x\|_{TV})(x_g)$ can be computed with cost $\mathcal{O}(p)$ [9]. However, the FISTA can not efficiently solve the composite regularization problem (1), since no efficient algorithm exists to solve the step

$$x^k = \arg\min_x \frac{1}{2}\|x - x_g\|^2 + \sum_{i=1}^m g_i(B_i x) \tag{4}$$

To solve (1), the key problem is thus to develop an efficient algorithm to solve (4). In the following section, we will show that a scheme based on composite splitting techniques can be used to do this.

3 Composite Splitting Algorithms

3.1 Problem Definition

We consider the following minimization problem:

$$min\{F(x) \equiv f(x) + \sum_{i=1}^{m} g_i(B_i x), x \in \mathbf{R^p}\} \tag{5}$$

where we make the following assumptions:

1. $g_i : \mathbf{R^p} \to \mathbf{R}$ is a continuous convex function for each $i \in \{1, \cdots, m\}$, which is possibly nonsmooth;
2. $f : \mathbf{R^p} \to \mathbf{R}$ is a smooth convex function of type $C^{1,1}$ and the continuously differential function with Lipschitz constant L_f: $\|\nabla f(x_1) - \nabla f(x_2)\| \leq L_f \|x_1 - x_2\|$ for every $x_1, x_2 \in \mathbf{R^p}$;
3. $\{B_i \in \mathbf{R}^{p \times p}\}_{i=1,...,m}$ are orthogonal matrices;
4. Problem (5) is solvable.

If $m = 1$, this problem will degenerate to problem (3) and may be efficiently solved by FISTA. However, it may be very hard to solve by ISTA/FISTA if $m > 1$. For example, we can suppose $m = 2$, $g_1(x) = ||x||_1$ and $g_2(x) = ||x||_{TV}$. When $g(x) = g_1(x)$ in the problem (3), the first step in Algorithm 2 has a closed form solution; When $g(x) = g_2(x)$ in the problem (3), the first step in Algorithm 2 can also be solved iteratively in a few iterations [9]. However, if $g(x) = g_1(x) + g_2(x)$ in (3), the first step in Algorithm 2 is not easily solved, which makes the computational complexity of each iteration so high that it is not practical to solve using FISTA.

When all function $\{g_i\}_{i=1,...,m}$ are smooth convex functions, this problem can be efficiently solved by the MSA/FaMSA. However, in our case, the function $\{g_i\}_{i=1,...,m}$ can be nonsmooth. Therefore, the MSA/FaMSA can not be directly applied to solve this problem. Of course, we may smooth these nonsmooth function first and then apply the FaMSA to solve it. However, in this case, the FaMSA with complexity bound $O(1/\sqrt{\epsilon})$ requires $O(1/\epsilon)$ iterations to compute an ϵ-optimal solution. It is obviously not optimal for the first order methods [24].

In the following, we propose our algorithm that overcomes these difficulties. Our algorithm decomposes the original problem (1) into m simpler regularization subproblems, where each of them is more easily solved by the FISTA.

3.2 Building Blocks

From the above introduction, we know that, if we can develop a fast algorithm to solve problem (4), the original composite regularization can then be efficiently solved by the FISTA, which obtains an ϵ-optimal solution in $\mathcal{O}(1/\sqrt{\epsilon})$ iterations. Actually, problem (4) can be considered as a denoising problem. We use composite splitting techniques to solve this problem: 1) splitting variable x into multiple variables $\{x_i\}_{i=1,...,m}$; 2) performing operator splitting over each of $\{x_i\}_{i=1,...,m}$ independently and 3) obtaining the solution x by linear combination of $\{x_i\}_{i=1,...,m}$. We call it Composite Splitting Denoising (CSD) method, which is outlined in Algorithm 3. Its validity is guaranteed by the following theorem:

Theorem 3. *Suppose $\{x^j\}$ the sequence generated by the CSD. If x^* is the true solution of problem (4), x^j will strongly converges to x^*.*

Due to page limitations, the proof for this theorem is given in the supplemental material.

Algorithm 3 CSD

```
Input: $\rho = 1/L$, $\alpha$, $\beta$, $\{z_i^0\}_{i=1,\ldots,m} = x_g$
for $j = 1$ to $J$ do
    for $i = 1$ to $m$ do
        $x_i = \arg\min_x \frac{1}{2m}\|x - z_i^{j-1}\|^2 + g_i(B_i x)$
    end for
    $x^j = \frac{1}{m}\sum_{i=1}^m x_i$
    for $i = 1$ to $m$ do
        $z_i^j = z_i^{j-1} + x^j - x_i$
    end for
end for
```

3.3 Composite Splitting Algorithm (CSA)

Combining the CSD with ISTA, a new algorithm, CSA, is proposed for composite regularization problem (5). In practice, we found that a small iteration number J in the CSD is enough for the CSA to obtain good reconstruction results. Especially, it is set as 1 in our algorithm. Numerous experimental results in the next section will show that it is good enough for real composite regularization problem.

Algorithm 4 outlines the proposed CSA. In each iteration, Algorithm 4 decomposes the original problem into m subproblems and solve them independently. For many problems in practice, these m subproblems are expected to be far easier to solve than the original joint problem. Another advantage of this algorithm is that the decomposed subproblems can be solved in parallel. Given x^{k-1}, the m subproblems to compute $\{y_i^k\}_{i=1,\cdots,m}$ are solved simultaneously in Algorithm 4.

3.4 Fast Composite Splitting Algorithms

In this section, a fast version of CSA named as FCSA is proposed to solve problem (5), which is outlined in Algorithm 5. FCSA decomposes the difficult composite regularization problem into multiple simpler subproblems and solve them in parallel. Each subproblems can be solved by the FISTA, which requires only $O(1/\sqrt{\epsilon})$ iterations to obtain an ϵ-optimal solution.

In this algorithm, if we remove the acceleration step by setting $t^{k+1} \equiv 1$ in each iteration, we will obtain the CSA. A key feature of the FCSA is its fast convergence performance borrowed from the FISTA. From Theorem 2, we know that the FISTA can obtain an ϵ-optimal solution in $\mathcal{O}(1/\sqrt{\epsilon})$ iterations.

Another key feature of the FCSA is that the cost of each iteration is $\mathcal{O}(mp\log(p))$, as confirmed by the following observations. The step $y_i^k = prox_\rho(g_i)(B_i(r^k - \frac{1}{L}\nabla f_i(r^k))$ can be computed with the cost $\mathcal{O}(p\log(p))$ for a lot of prior models g_i. The step

Algorithm 4 CSA

```
Input: $\rho = 1/L$, $x^0$
repeat
  for $k = 1$ to $K$ do
    for $i = 1$ to $m$ do
      $y_i^k = prox_\rho(g_i)(B_i(x^{k-1} - \frac{1}{L}\nabla f_i(x^{k-1}))$
    end for
    $x^k = \frac{1}{m}\sum_{i=1}^m B_i^{-1} y_i^k$
  end for
until Stop criterions
```

Algorithm 5 FCSA

```
Input: $\rho = 1/L$, $t^1 = 1$ $x^0 = r^1$
repeat
  for $k = 1$ to $K$ do
    for $i = 1$ to $m$ do
      $y_i^k = prox_\rho(g_i)(B_i(r^k - \frac{1}{L}\nabla f_i(r^k))$
    end for
    $x^k = \frac{1}{m}\sum_{i=1}^m B_i^{-1} y_i^k$
    $t^{k+1} = \frac{1+\sqrt{1+4(t^k)^2}}{2}$
    $r^{k+1} = x^k + \frac{t^k - 1}{t^{k+1}}(x^k - x^{k-1})$
  end for
until Stop criterions
```

$x^k = \frac{1}{m}\sum_{i=1}^m B_i^{-1} y_i^k$ can also be computed with the cost of $\mathcal{O}(p\log(p))$. Other steps only involve adding vectors or scalars, thus cost only $\mathcal{O}(p)$ or $\mathcal{O}(1)$. Therefore, the total cost of each iteration in the FCSA is $\mathcal{O}(mp\log(p))$.

With these two key features, the FCSA efficiently solves the composite regularization problem (5) and obtains better results in terms of both the accuracy and computation complexity. The experimental results in the next section demonstrate its superior performance.

4 Experiments

4.1 Application on MR Image Reconstruction

Specifically, we apply the CSA and FCSA to solve the Magnetic Resonance (MR) image recovery problem in compressive sensing [1]:

$$\min_x F(x) \equiv \frac{1}{2}\|Ax - b\|^2 + \alpha\|\Phi^{-1}x\|_{TV} + \beta\|x\|_1 \quad (6)$$

where $A = R\Phi^{-1}$, R is a partial Fourier transform, Φ^{-1} is the wavelet transform, b is the under-sampled Fourier measurements, α and β are two positive parameters.

This model has been shown to be one of the most powerful models for the compressed MR image recovery [1]. However, since the $\|\Phi^{-1}x\|_{TV}$ and $\|x\|_1$ are both nonsmooth in x, this problem is much more difficult to solve than any of those with a single nonsmooth term such as the L1 regularization problem or a total variation regularization problem. In this case, the FISTA can efficiently solve the L1 regularization problem [8], since the first step $x^k = prox_\rho(\|x\|_1)(r^k - \rho \nabla f(r^k))$ has a close form solution in Algorithm 2. The FISTA can also efficiently solve the total variation regularization problem [9], since the first step $x^k = prox_\rho(\|\Phi^{-1}x\|_{TV})(r^k - \rho \nabla f(r^k))$ can be computed quickly in Algorithm 2. However, the FISTA can not efficiently solve the joint L1 and TV regularization problem (6), since $x^k = prox_\rho(\alpha\|\Phi^{-1}x\|_{TV} + \beta\|x\|_1)(r^k - \rho \nabla f(r^k))$ can not be computed in a short time.

The Conjugate Gradient (CG) [1] has been applied to the problem (6) and it converges very slowly. The computational complexity has been the bottleneck that made (6) impractical in the past [1]. To use this model for practical MR image reconstruction, Ma et al. proposed a fast algorithm based on the operator splitting technique [2], which is called TVCMRI. In [3], a variable splitting method (RecPF) was proposed to solve the MR image reconstruction problem. Both of them can replace iterative linear solvers with Fourier domain computations, which can gain substantial time savings. To our knowledge, they are two of the fastest algorithms to solve problem (6) so far. Different from their algorithms, the CSA and FCSA directly attack the joint L1 and total variation norm regularization problem by transferring it to the L1 regularization and TV norm regularization subproblems, which can be efficiently solved. In the following, we compare our CSA and FCSA with their algorithms. The results show that the FCSA is far more efficient than the TVCMRI and RecPF.

Experiment Setup Suppose a MR image x has n pixels, the partial Fourier transform R in problem (6) consists of m rows of a $n \times n$ matrix corresponding to the full 2D discrete Fourier transform. The m selected rows correspond to the acquired b. The sampling ratio is defined as m/n. The scanning duration is shorter if the sampling ratio is smaller. In MR imaging, we have certain freedom to select rows, which correspond to certain frequencies. In the k-space, we randomly obtain more samples in low frequencies and less samples in higher frequencies. This sample scheme has been widely used for compressed MR image reconstruction [1–3]. Practically, the sampling scheme and speed in MR imaging also depend on the physical and physiological limitations [1].

All experiments are conducted on a 2.4GHz PC in Matlab environment. We compare the CSA and FCSA with two of the fastest MR image reconstruction methods, TVCMRI [2] and RecPF [3]. For fair comparisons, we download the codes from their websites and carefully follow their experiment setup. For example, the observation measurement b is synthesized as $b = Rx + \mathbf{n}$, where $\mathbf{n}$ is the Gaussian white noise with standard deviation $\sigma = 0.01$. The regularization parameter α and β are set as 0.001 and 0.035. R and b are given as inputs, and x is the unknown target. For quantitative evaluation, we compute the Signal-to-Noise Ratio (SNR) for each reconstruction result.

Numerical Results We perform experiments on a full body MR image with size of 924×208. Each algorithm runs 50 iterations. The sample ratio is set to be approximately 25%. To reduce the randomness, we run each experiments 100 times for each parameter setting of each method. Due to page limitations, we include the experimental results and comparisons in the supplemental materials. The examples of the original and recovered images by different algorithms are shown in Figure 1. From there, we can observe that the results obtained by the FCSA are not only visibly better, but also superior in terms of both the SNR and CPU time.

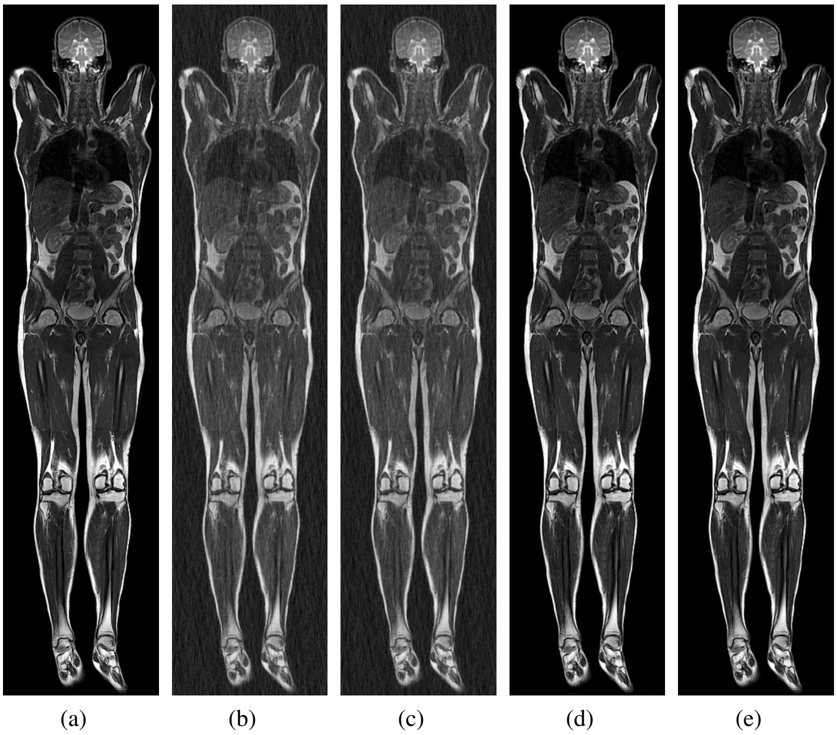
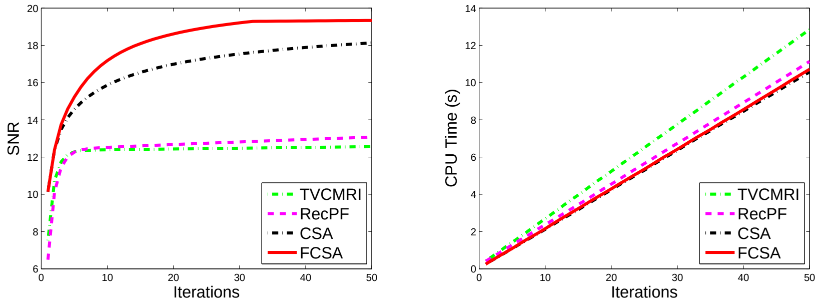


Fig. 1. Full Body MR image reconstruction from 25% sampling (a) Original image; (b), (c), (d) and (e) are the reconstructed images by the TVCMRI [2], RecPF [3], CSA and FCSA. Their SNR are 12.56, 13.06, 18.21 and 19.45 (db). Their CPU time are 12.57, 11.14, 10.20 and 10.64 (s).

To further evaluate the reconstruction performance, we use sampling ratio 25% to obtain the measurement b. Different methods are then used to perform reconstruction. To reduce the randomness, we run each experiments 100 times for each parameter setting of each method. The SNR and CPU time are traced in each iteration for each methods. Figure 2 gives the performance comparisons between different methods in terms of the CPU time and SNR. The reconstruction results produced by the FCSA are far better than those produced by the CG, TVCMRI and RecPF. The reconstruction performance of the FCSA is always the best in terms of both the reconstruction accuracy and the computational complexity, which further demonstrate the effectiveness and efficiency of the FCSA for the compressed MR image construction.



(a) (b)

Fig. 2. Performance comparisons with sampling ratio 25%: a) Iterations vs. SNR (db) and (b) Iterations vs. CPU Time (s).

4.2 Application on Low-Rank Tensor Completion

We also apply the the proposed FCSA to the low rank tensor completion problem. This problem has gained a lot of attentions recently [25, 26, 10, 27]. It is formulated as follows:

$$\min_X F(X) \equiv \frac{1}{2}\|\mathcal{A}(X) - b\|^2 + \alpha\|X\|_* \tag{7}$$

where $X \in \mathbb{R}^{p \times q}$ is a unknown matrix, $\mathcal{A} : \mathbb{R}^{p\times q} \to \mathbb{R}^n$ is the linear map, and $b \in \mathbb{R}^n$ is the observation. The nuclear norm is defined as $\|X\|_* = \sum_i \sigma_i(X)$, where $\sigma_i(X)$ is the singular value of the matrix X. The accelerated proximal gradient (APG) scheme in the FISTA has been used to solve (7) in [10]. In most cases, the APG gains the best performance compared with other methods, since it can obtain an ϵ-optimal solution in $O(1/\sqrt{\epsilon})$ iterations.

Similarly, the tensor completion problem can be defined. We use the 3-mode tensor as an example for the low rank tensor completion. It is easy to extend to the n-mode tensor completion. The 3-mode tensor completion can be formulated as follows [28]:

$$\min_{\mathcal{X}} F(\mathcal{X}) \equiv \frac{1}{2}\|\mathcal{A}(\mathcal{X}) - b\|^2 + \sum_{i=1}^{m} \alpha_i \|B_i \mathcal{X}\|_* \tag{8}$$

where $\mathcal{X} \in \mathbb{R}^{p\times q\times m}$ is the unknown 3-mode tensor, $\mathcal{A} : \mathbb{R}^{p\times q\times m} \to \mathbb{R}^n$ is the linear map, and $b \in \mathbb{R}^n$ is the observation. B_1 is the "unfold" operation along the 1-mode on a tensor $\mathcal{X}$, which is defined as $B_1\mathcal{X} := X_{(1)} \in \mathbb{R}^{p\times qm}$; B_2 is the "unfold" operation along the 2-mode on a tensor $\mathcal{X}$, which is defined as $B_2\mathcal{X} := X_{(2)} \in \mathbb{R}^{q\times pm}$; B_3 is the "unfold" operation along the 3-mode on a tensor $\mathcal{X}$, which is defined as $B_3\mathcal{X} := X_{(3)} \in \mathbb{R}^{m\times pq}$. The opposite operation "fold" is defined as $B_i^T X_i = \mathcal{X}$ where $i = 1, 2, 3$.

Generally, it is far harder to solve the tensor completion problem than the matrix completion because of the composite regularization. The solvers in [10] can not be used

Table 1. Comparisons of the CPU Time and RSE

	CGD-LRTC [28]		**APG-LRMC** [10]		**FCSA-LRTC**	
	Time (s)	RSE	Time (s)	RSE	Time (s)	RSE
Window	133.21	0.3843	100.98	0.0962	133.56	0.0563
Cherry	134.39	0.5583	102.43	0.3201	134.65	0.1069
Sheep	134.96	0.5190	101.33	0.1784	131.23	0.1017
Fish	136.29	0.5886	99.89	0.2234	135.31	0.1056

to efficiently solve (8). In [28], a relaxation technique is used to separate the dependant relationships and the block coordinate descent (BCD) method is used to solve the low rank tensor completion problem. As far as we know, it is the best method for the low rank tensor completion so far. However, it converges very slow due to the convergence properties of the BCD. Fortunately, the proposed FCSA can be directly used to efficiently solve 8. Different from the BCD method for LRTC using relaxation techniques [28], the FCSA can directly attack the composite matrix nuclear norm regularization problem by transforming it to multiple matrix nuclear norm regularization subproblems, which can be efficiently solved in parallel. In the following, we compare the proposed FCSA and BCD for the low rank tensor completion. We called them FCSA-LRTC and CBD-LRTC respectively. The results show that the FCSA is far more efficient than the BCD for the LRTC problem.

Experiment Setup Suppose a color image $\mathcal{X}$ with low rank has the size of $h \times w \times d$, where h, w, d denote its height, width and color channel respectively. When the color values of some pixels are missing in the image, the tensor completion is conducted to recover the missed values. Suppose q pixels miss the color values in the image, the sampling ratio is defined as $(h \times w \times d - q \times d)/(h \times w \times d)$. The known color values in the image are called the samples for tensor completion. We randomly obtain these samples or designate the samples before the tensor completion [28].

All experiments are conducted on a 2.4GHz PC in Matlab environment. We compare the proposed FCSA-LRTC with the CGD-LRTC [28] for the tensor completion problem. To show the advantage of the LRTC over the low rank matrix completion (LRMC), we also compare the proposed FCSA-LRTC with the APG based LRMC method (APG-LRMC)[10]. As introduced in the above section, the APG-LRMC is not able to solve the tensor completion problem (8) directly. For comparisons, we approximately solve (8) by using the APG-LRMC to conduct the LRMC in d color channels independently. For quantitative evaluation, we compute the Relative Square Error (RSE) for each completion result. The RSE is defined as $\|\mathcal{X}_c - \mathcal{X}\|_F / \|\mathcal{X}\|_F$, where $\mathcal{X}_c$ and $\mathcal{X}$ are the completed image and ground-truth image respectively.

Numerical Results We apply different methods on four 2D color images respectively. To perform fair comparisons, all methods run 50 iterations. Figure 3 shows the visual comparisons of the completion results. In this case, the visual effects obtained by the FCSA-LRTC are also far better than those of the CGD-LRTC [28] and slightly better than those obtained by the APG-LRMC [10]. Table 1 tabulates the RSE and CPU Time

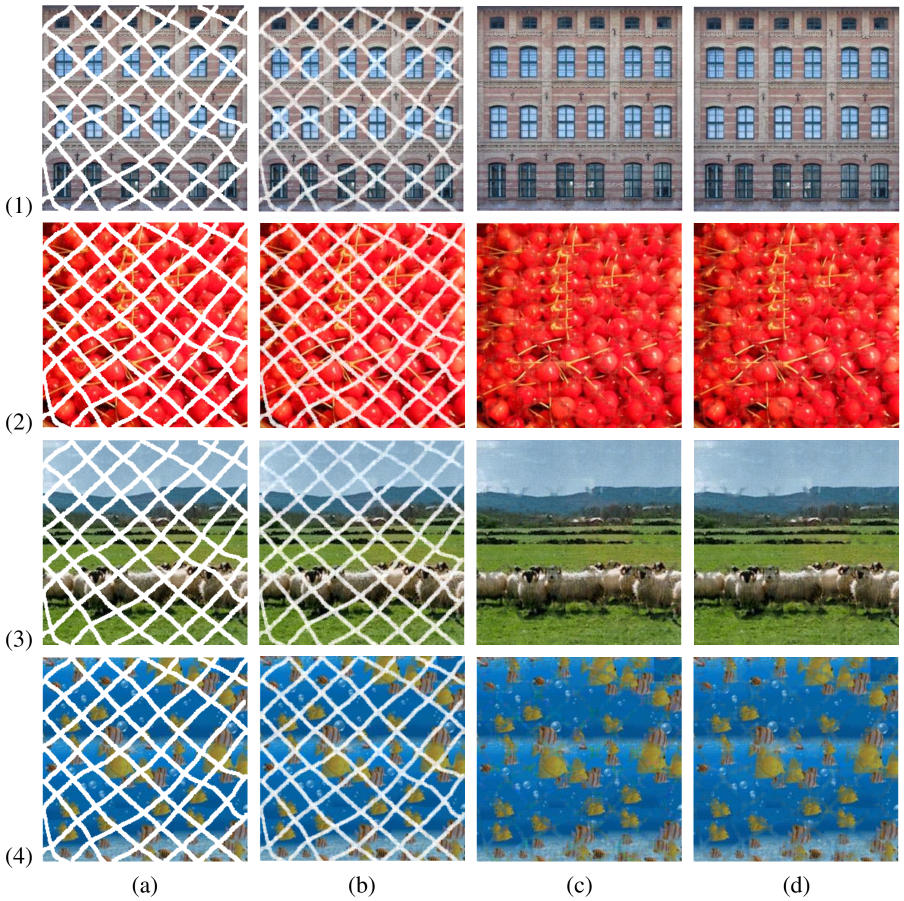


Fig. 3. Comparisons in terms of visual effects. Color images are: (1) Window; (2) Cherry; (3) Sheep and (4) Fish. The column (a), (b), (c) and (d) correspond to the images before completion, the obtained results by the CGD-LRTC [28], APG-LRMC [10] and FCSA-LRTC, respectively.

by different methods on different color images. The FCSA-LRTC always obtains the smallest RSE in all color images, which shows its good performance for the low rank tensor completion.

5 Conclusion

In this paper, we proposed composite splitting algorithms based on splitting techniques and optimal gradient techniques for the mixture prior model optimization. The proposed algorithms decompose a hard composite regularization problem into multiple simpler subproblems and efficiently solve them in parallel. This is very attractive for practical applications involving large-scale data optimization. The computation complexities of the proposed algorithms are very low in each iteration. The promising numerical results on applications of compressed MR image reconstruction and low-rank tensor completion validate the advantages of the proposed algorithms.

References

1. M. Lustig, D.D., Pauly, J.: Sparse mri: The application of compressed sensing for rapid mr imaging. In: Magnetic Resonance in Medicine. (2007)
2. Ma, S., Yin, W., Zhang, Y., Chakraborty, A.: An efficient algorithm for compressed mr imaging using total variation and wavelets. In: Proceedings of CVPR. (2008)
3. Yang, J., Zhang, Y., Yin, W.: A fast alternating direction method for tvl1-l2 signal reconstruction from partial fourier data. IEEE Journal of Selected Topics in Signal Processing, Special Issue on Compressive Sensing **4** (2010)
4. Gabay, D.: Applications of the method of multipliers to variational inequalities. In: Augmented Lagrange Methods: Applications to the Solution of Boundary-valued Problems, North Holland, Amsterdam (1983) 299–331
5. Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. SIAM Journal on Multiscale Modeling and Simulation **19** (2008) 1107–1130
6. Tseng, P.: A modified forward-backward splitting method for maximal monotone mappings. SIAM Journal on Control and Optimization **38** (2000) 431–446
7. Hale, E.T., Yin, W., Zhang, Y.: Fixed-point continuation for .1-minimization: Methodology and convergence. SIAM Journal on Optimization **19** (2008) 1107–1130
8. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences **2** (2009) 183–202
9. Beck, A., Teboulle, M.: Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Transaction on Image Processing **18** (2009) 2419–2434
10. Toh, K., Yun, S.: An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. Pacific Journal of Optimization (2009)
11. Ji, S., Ye, J.: An accelerated gradient method for trace norm minimization. In: Proceedings of ICML. (2009)
12. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite-element approximations. Computers and Mathematics with Applications **2** (1976) 17–40
13. Glowinski, R., Tallec, P.L.: Augmented lagrangian and operator-splitting methods in nonlinear mechanics. SIAM (1989)

14. Tseng, P.: Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. SIAM Journal on Control and Optimization **29** (1991) 119–138
15. He, B.S., Liao, L.Z., Han, D., Yang, H.: A new inexact alternating direction method for monotone variational inequalities. Mathematical Programming **92** (2002) 103–118
16. Candes, E.J., Romberg, J., Tao, T.: Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on Information Theory **52** (2006) 489–509
17. Donoho, D.: Compressed sensing. IEEE Trans. on Information Theory **52** (2006) 1289–1306
18. Goldstein, T., Osher, S.: The split bregman algorithm for l1 regularized problems. Technical report, UCLA, CAM Report 08-29 (2008)
19. Afonso, M., Bioucas-Dias, J., Figueiredo, M.: Fast image recovery using variable splitting and constrained optimization. In: IEEE Transactions on Image Processing, Submitted (2009)
20. Yang, J., Zhang, Y.: Alternating direction algorithms for l1-problems in compressive sensing. Technical report, Rice University, TR 09-37 (2009)
21. Yuan, X.: Alternating direction methods for sparse covariance selection, preprint (2009)
22. Goldfarb, D., Ma, S.: Fast multiple splitting algorithms for convex optimization. Technical report, Department of IEOR, Columbia University, New York (2009)
23. Nesterov, Y.E.: A method for solving the convex programming problem with convergence rate o$(1/k^2)$. Dokl. Akad. Nauk SSSR **269** (1983) 543–547
24. Nesterov, Y.E.: Gradient methods for minimizing composite objective function. Technical report, http://www.ecore.beDPs/dp1191313936.pdf (2007)
25. Ma, S., Goldfarb, D., Chen, L.: Fixed point and bregman iterative methods for matrix rank minimization. Mathematical Programming (2010)
26. Cai, J.F., Candes, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization **20** (2010) 1956–1982
27. Yang, J., Yuan, X.: An inexact alternating direction method for trace norm regularized least squares problem. In: Preprint. (2010)
28. Liu, J., Musialski, P., Wonka, P., Ye, J.: Tensor completion for estimating missing values in visual data. In: Proceedings of ICCV. (2009)