Convergence of an alternating direction and projection method for sparse dictionary learning

Yijia Zhou¹ and Lijun Xu², ³
¹School of Computer & Software, Dalian Neusoft University of Information, Dalian 116023, Liaoning, China
²School of Science, Dalian Maritime University, Dalian 116026, Liaoning, China
³Email: lijun_xu@dlmu.edu.cn

Abstract. In this paper, we focus on sparse dictionary learning that is widely used as a data processing technique in many real-world applications. Based on the frame of alternating direction method of multiplier (ADMM), we extend to an alternating direction and projection method for sparse dictionary learning. By introducing proximal mapping and the equivalence to the corresponding projection, a partial convergence result of this multi-block and nonconvex ADMM algorithm is given that the algorithm converges to a Karush-Kuhn-Tucker point whenever it converges.

1. Introduction
Dictionary learning, a data processing technique widely used in many applications including signal processing, compressive sensing and machine learning. The research in dictionary learning has followed three main directions that correspond to three categories of algorithms: 1) the probabilistic learning methods; 2) the learning methods based on clustering or vector quantization; and 3) the methods for learning dictionaries with a particular construction. This construction in 3) is typically driven by priors on the structure of the data or to the target usage of the learned dictionary.

Dictionary learning is to decompose a sampled data matrix Y into a product of two factors, say D and X where D is called a dictionary and X is a representation of the data (or a coding matrix) under the dictionary. As usual, some desired properties, such as nonnegativity and sparsity, can be imposed on either or both factors. In recent years, dictionary learning for sparse representation (or say sparse dictionary learning) has found successful applications in real world. For example, it has been applied to medical imaging and representation of signals, audio and visual data. Designing dictionaries to better fit the above decomposition can be done by either selecting one from a pre-specified set of linear transforms, or by adapting the dictionary to a set of training signals. Both these techniques have been considered, but this topic is largely still open.

1.1. Sparse dictionary learning
In recent years, there is a growing interest in the study of dictionary learning for sparse representations of real data, like signals, medical images, audios and visuals. Roughly speaking, we say that a signal \( y \in \mathbb{R}^n \) admits a sparse representation under a dictionary \( D \in \mathbb{R}^{m \times n} \) if one can find a linear combination of “a few” columns (atoms) from \( D \) that is “close” to the signal \( y \). Sparse representations serve useful purposes in many data processing tasks, and a key to success is to have a sufficiently good dictionary. In many situations, a good dictionary, such as a wavelet basis, is known a priori. Most of
earlier works in this field have been done based on this premise, and algorithms have been developed to reconstruct signals from a given dictionary and associated measurements, such algorithms [1-6]. More recently, there is a growing body of works without assuming that a dictionary is known. Instead, a dictionary is constructed for training data using learned techniques such as [7-12], for example. Under favorable conditions, learning the dictionary instead of using off-the-shelf bases has been shown to dramatically improve data reconstruction.

Again, most algorithms in dictionary learning are developed based on minimizing data fidelity functions either with penalty/regularization terms or with explicit constraints (occasionally with both). For instance, the popular algorithm K-SVD [11] widely used to learn dictionaries for sparse representation, is built on minimizing the Frobenius-norm data fidelity with explicit sparsity constraints. For very large training sets and dynamic (time-dependent) training data, a number of approximate matrix factorization models and so-called online algorithms have been proposed that also enforce sparsity structures by various means (see [13-15], for example). In this paper, we focus on the sparse dictionary learning with explicit sparsity constraints from the view point of matrix factorization. In addition, there are also some works on Lagrange multiplier in various fields including the variational theory (see [16-19], for example) and optimization theory. We focus on a kind of Lagrange multiplier method by means of optimization framework in this paper.

1.2. Matrix factorization model of sparse dictionary learning
Sparse dictionary learning can be formulated as a specific matrix factorization problem. As in [20], we denote a training dataset by Y, a dictionary by D and a sparse representation by X. The concerned sparse dictionary learning model takes the following form,

$$\min_{D,X} ||Y - DX||^2_F \quad \text{s.t.} \quad D \in \mathcal{D}, \ X \in \mathcal{X},$$

(1)

where,

$$\mathcal{D} = \{d_1, \cdots, d_p\} \in \mathbb{R}^{m \times p}; \ |d_i|_2 = 1, \ \forall i = 1, \cdots, p,$$

$$\mathcal{X} = \{x_1, \cdots, x_n\} \in \mathbb{R}^{p \times n}; \ |x_i|_0 \leq K, \ \forall i = 1, \cdots, n.$$

The constraint both $\mathcal{D}$ and $\mathcal{X}$ are easily projectable sets so that we can apply the class of projection algorithm without difficulty.

1.3. The main contribution
We extend the classic alternating direction method of multipliers (ADMM) to an alternating direction and projection method for solving (1). By introducing proximal mapping of indicator function and inducing the equivalence to the corresponding projection operator, we give a partial convergence result of this multi-block and nonconvex ADMM algorithm. That is, the algorithm converges to a Karush-Kuhn-Tucker point whenever it converges. This provides some assurance on the behavior of the ADMM-type algorithm applied to sparse dictionary learning problem.

This paper is organized as follows. In Sect. 2, we introduce the classic ADMM and propose an alternating direction and projection method for (1). Sect. 3 contains a convergence result for the proposed method. A numerical example is tested in Sect. 4. Finally, we conclude this paper in Sect. 5.

2. An alternating direction and projection method for sparse dictionary learning
In this section, we first introduce the classic ADMM, then deduce a projection method based on ADMM algorithmic frame for (1) according to the previous work [20].

2.1. Classic ADMM method
In a finite-dimensional setting, the classic alternating direction method of multipliers (ADMM or simply ADM) is designed for solving separable convex programs of the form

$$\min_{x \in \mathcal{X}, y \in \mathcal{Y}} f(x) + g(y) \quad \text{s.t.} \quad Ax + By = c,$$

(2)

where $f$ and $g$ are convex functions defined on closed convex subsets $\mathcal{X}$ and $\mathcal{Y}$ of finite-dimensional spaces, respectively, and $A, B$ and $c$ are matrices and vector of appropriate sizes. The augmented Lagrangian function of (2) is
\[ \mathcal{L}_A(x, y, \lambda) = f(x) + g(y) + \lambda^T (Ax + By - c) + \frac{\beta}{2} \| Ax + By - c \|_2^2, \]  
where \( \lambda \) represents a Lagrangian multiplier vector and \( \beta > 0 \) is a penalty parameter.

ADMM method [21, 22] is an extension of the classic augmented Lagrange multiplier method. It performs one sweep of alternating minimization with respect to \( x \) and \( y \) individually, then updates the multiplier \( \lambda \). At the \( k \)-th iteration, an ADMM scheme executes the following steps: given \((x^k, y^k, \lambda^k)\),

\[
\begin{align*}
    x^{k+1} &\leftarrow \arg\min_{x \in \mathcal{X}} \mathcal{L}_A(x, y^k, \lambda^k), \\
y^{k+1} &\leftarrow \arg\min_{y \in \mathcal{Y}} \mathcal{L}_A(x^{k+1}, y, \lambda^k), \\
    \lambda^{k+1} &\leftarrow \lambda^k + \gamma B (Ax^{k+1} + By^{k+1} - c),
\end{align*}
\]

where \( \gamma \in (0,1.618) \) is step length. It is worth noting that (4) only involves \( f(x) \) in the objective and (5) only \( g(y) \), whereas the classic augmented Lagrangian multiplier method requires a joint minimization with respect to both \( x \) and \( y \), that is, substituting steps (4) and (5) by

\[(x^{k+1}, y^{k+1}) \leftarrow \arg\min_{x \in \mathcal{X}, y \in \mathcal{Y}} \mathcal{L}_A(x, y, \lambda^k),\]

which involves both \( f(x) \) and \( g(y) \) and is usually more difficult to solve.

2.2. Extension with projection to sparse dictionary learning

As in [20], to facilitate an efficient use of alternating minimization, we introduce two auxiliary variables \( U \in \mathbb{R}^{m \times p} \) and \( V \in \mathbb{R}^{p \times n} \) and consider the following model equivalent to (1),

\[
\begin{align*}
    \min_{D, X, U, Y} &\frac{1}{2} \| Y - DX \|_F^2 \text{ s.t. } D - U = 0, \ X - V = 0, \ U \in \mathcal{D}, \ V \in \mathcal{X}.
\end{align*}
\]

The augmented Lagrangian function of (7) is

\[
\mathcal{L}_A(D, X, U, V, \Lambda, \Pi) = \frac{1}{2} \| Y - DX \|_F^2 + \Lambda \cdot (D - U) + \Pi \cdot (X - V) + \frac{\alpha}{2} \| D - U \|_F^2 + \frac{\beta}{2} \| X - V \|_F^2,
\]

where \( \Lambda \in \mathbb{R}^{m \times p}, \Pi \in \mathbb{R}^{p \times n} \) are Lagrangian multipliers and \( \alpha, \beta > 0 \) are penalty parameters for the constraints \( X - U = 0 \) and \( Y - V = 0 \), respectively, and the scalar product “\( \cdot \)” of two equal-size matrices \( A \) and \( B \) is the sum of all element-wise products, i.e., \( A \cdot B = \text{trace} (A^T B) = \sum_{i,j} a_{ij} b_{ij} \).

The alternating direction method of multipliers (ADMM) for (8) is derived by successively minimizing the augmented Lagrangian function \( \mathcal{L}_A \) with respect to \( D, X \) and \( (U, V) \), one at a time while fixing others at their most recent values, and then updating the multipliers after each sweep of such alternating minimization. The introduction of the two auxiliary variables \( U \) and \( V \) makes it easy to carry out each of the alternating minimization steps. Specifically, these steps can be written in the following closed form,

\[
\begin{align*}
    D_+ &= (YX^T + \alpha U - \Lambda)(XX^T + \alpha I)^{-1}, \tag{9} \\
    X_+ &= (D_+^T D_+ + \beta I)^{-1}(D_+^T Y + \beta V - \Pi), \tag{10} \\
    U_+ &= \mathcal{P}_D(D_+ + \Lambda/\alpha), \tag{11} \\
    V_+ &= \mathcal{P}_\mathcal{X}(X_+ + \Pi/\beta), \tag{12} \\
    \Lambda_+ &= \Lambda + \alpha(D_+ - U_+), \tag{13} \\
    \Pi_+ &= \Pi + \beta (X_+ - V_+), \tag{14}
\end{align*}
\]

where \( \mathcal{P}_\mathcal{X} (\mathcal{P}_D) \) stands for the projection onto the set \( \mathcal{X} (\mathcal{D}) \) in Frobenius norm, and the subscript “\( + \)” is used to denote values at the new iteration. We stop the updating (9-14) either when data fidelity does not change or both variables \( D \) and \( X \) do not change meaningfully in several consecutive iterations, for given tolerance \( tol > 0 \),

\[ \min \{ |f - f_k|/|f|, \max(\|D - D_k\|/\|D\|, \|X - X_k\|/\|X\|) \} \leq tol. \]

Since the involved inverse matrices are both \( p \times p \), which is usually small in dictionary learning, the linear systems (9) and (10) are relatively inexpensive, especially for \( p \ll \max(m, n) \). On the other hand, when \( p \) is relatively large as considered in [20], instead of using the inversions in (9) and (10), it will be more efficient to employ suitable iterative procedures, such as the conjugate gradient method, to approximately solve the two convex quadratic minimization problems in (4) and (5).

We note that the similar algorithms with fixed and adaptive penalty parameters, has been studied in [23] for NMF and in [20] for structured-enforced matrix factorization. Moreover, [23] gives a
convergence of the algorithm for NMF which possess convex nonnegative projections. [20] consider a unified structure-enforced matrix factorization which contains sparse dictionary learning, but there are no theoretical results since complexity of nonconvex structures and object function. The current work provides convergence of the extended algorithm for sparse dictionary learning which is several meaningful and nontrivial extensions beyond the work in [23] and [20].

3. Convergence to KKT points

It is known that the classic ADMM methodology has theoretical guarantees of convergence for convex programs of two separable variables (see [24–26], for example). However, the convergence of 3-block ADMM is generally not guaranteed [27]. Some other sufficient conditions ensuring the convergence of multi-block ADMM were introduced, see [28–32] for example. In spite of the success of ADMM on convex problems, the behavior of ADMM on nonconvex problems has been largely a mystery, especially when there are nonconvex, non-separable and nonsmoothed functions in the problem (1).

In this section, we provide a partial result on the convergence of the proposed algorithm, i.e., (9-14). Denote as the indicate function of the set \( \mathcal{X} \), namely when \( x \in \mathcal{X} \), \( \delta_x(x) = 0 \) and \( \delta_x(x) = +\infty \) otherwise. Let \( U_i \) be the i-th column of \( U \), since the normalization set \( \mathcal{D} \) has explicit formulation, we can rewrite (7) to the following form,

\[
\begin{align*}
\min_{D,X,U,V} & \quad \frac{1}{2} \left\| Y - DX \right\|_F^2 + \delta_X(V) \\
\text{s.t.} & \quad D - U = 0, \quad X - V = 0, \\
& \quad U_i^T U_i - 1 = 0, i = 1, \ldots, k.
\end{align*}
\]

(15)

To simplify notation, let us define \( Z \equiv (D,X,U,V) \), then a point \( Z \) is a KKT point (see [24–32] for example) of problem (7) and (15) if it satisfies the KKT conditions with proper \( \Lambda, \Pi \) and \( \lambda \):

\[
\begin{align*}
(DX - Y)^T + \Lambda = 0, \\
D^T(DX - Y) + \Pi = 0, \\
D - U = 0, \\
X - V = 0, \\
U_i^T U_i - 1 = 0, i = 1, \ldots, k, \\
\text{diag}(\lambda) U - \Lambda = 0, \\
\Pi \in \partial \delta_X(V), \\
V \in \mathcal{X},
\end{align*}
\]

(16) (17) (18) (19) (20) (21) (22) (23)

where \( \text{diag}(\lambda) \) is a \( k \times k \) diagonal matrix with diagonal elements \( \{\lambda_i\}_{i=1}^k \), namely Lagrange multipliers of \( U_i^T U_i - 1 = 0 \). \( \partial \delta_X(V) \) is limit subdifferential of the indicate function \( \delta_X(\cdot) \) at \( V \).

3.1. Proximal mapping

Since the set \( \mathcal{X} \) represents each column of \( V \in \mathcal{X} \) is sparse which is easily projected, we interpret its subdifferential (22) from the view of proximal mapping. Consider the proximal mapping of \( \delta_X(\cdot) \):

\[
\text{prox}_{\beta \delta_X} (z) := \arg \min_{\hat{z}} \left\{ \frac{\beta}{2} \| z - x \|^2 + \delta_X(\hat{z}) \right\}.
\]

Let \( z^* = \text{prox}_{\beta \delta_X} (x) \), according to the first order necessary conditions, we have

\[
0 \in \beta (z^* - x) + \partial \delta_X (z^*).
\]

Therefore, (22) can be equivalent to

\[
V = \text{prox}_{\beta \delta_X} (V + \Pi / \beta).
\]

(24)

Since the proximal mapping of an indicate function of the closed and nonempty set is actually the projection operator on the set, we can obtain

\[
V = P_{\mathcal{X}} (V + \Pi / \beta).
\]

(25)

Denote \( J = \{(i,j) : V_{ij} \neq 0, 1 \leq i \leq k, 1 \leq j \leq n\} \) as the index of nonzeros in \( V \). It’s obviously obtained that \( (V + \Pi / \beta)_{j} = V_j \), then \( \Pi_j = 0 \). Thus, the complementary condition is also satisfied.
3.2. Convergence analysis

Simply denote $Z_k \triangleq (D_k, X_k, U_k, V_k, \Lambda_k, \Pi_k)$ as the $k$-th iterative pint of updating rule (9-14), then we can obtain the following convergence result.

**Theorem 1** Let $\{Z_k\}_{k=1}^{\infty}$ be a sequence generated by the algorithm (9-14) with fixed penalties that satisfies the condition

$$\lim_{k \to \infty} (Z_{k+1} - Z_k) = 0.$$  

Then any accumulation point of $\{D_k, X_k, U_k, V_k\}_{k=1}^{\infty}$ is a KKT point of (7). Consequently, any accumulation point of $\{D_k, X_k\}_{k=1}^{\infty}$ is a KKT point of (1).

**Proof.** We can rearrange the update formulas (9-14) into

$$(D_k - D)(XX^T + \alpha I) = -(D^T X - Y)X^T + \alpha(D - U) + \Lambda),$$

$$(D_k^T D_k + \beta I)(X_k - X) = -(D_k^T (D_k - U) + \beta(X - X) + \Pi),$$

$$U_k - U = \mathcal{P}_D(D_k + \Lambda_k/\alpha) - U,$$

$$V_k - V = \mathcal{P}_\mathcal{X}(X_k + \Pi_k/\beta) - V,$$

$$\Lambda_k - \Lambda = \alpha(D_k - U_k),$$

$$\Pi_k - \Pi = \beta(X_k - V_k).$$

The assumption $Z_k - Z \to 0$ implies that both sides above all go to zero. Thus, after adding subscript $k$ to all variables $D, X, \cdots$ replacing “+”, letting $k \to +\infty$, we have

$$(D_k - D)XX^T + \Lambda_k \to 0,$$  

$$(D_k^T D_k + \beta I)(X_k - X) \to 0,$$  

$$\mathcal{P}_D(D_k + \Lambda_k/\alpha) - U_k \to 0,$$  

$$\mathcal{P}_\mathcal{X}(X_k + \Pi_k/\beta) - V_k \to 0,$$  

$$D_k - U_k \to 0,$$  

$$X_k - V_k \to 0,$$  

where $(D_k - U_k)$ and $(X_k - V_k)$ in (27) and (28) have been eliminated from (31) and (32). Clearly, the first four equations in the KKT conditions (16-19) are satisfied at any limit point $Z \triangleq (\bar{D}, \bar{X}, \bar{U}, \bar{V}, \bar{\Lambda}, \bar{\Pi}).$

The normalization of $\bar{D}$ and sparsity of $\bar{V}$ are guaranteed by the updating rule. Thus, we only need to verify (21) and (22). Now we examine the following two equations (29) and (30) at $\bar{Z}$,

$$\mathcal{P}_D (\bar{D} + \bar{\Lambda}/\alpha) = \bar{U},$$  

$$\mathcal{P}_\mathcal{X} (\bar{X} + \bar{\Pi}/\beta) = \bar{V}.$$  

According to the above derivation from (22) to (25), (34) is also equivalent to the form of (22). Since $\mathcal{D} = \{d_l, \cdots, d_k\}: \|d_i\|_2 = 1, \forall i\}$ and $\bar{D} = \bar{U} \in \mathcal{D}$, then from (33) we can get $(\bar{D} + \bar{\Lambda}/\alpha)_j = \bar{D}_j + \bar{\Lambda}_j/\alpha = \bar{\Lambda}_j/\alpha (\bar{\Lambda}_j \neq 0)$, namely, $\bar{\Lambda}_j = \alpha(\bar{\Lambda}_j - 1)\bar{D}_j$. Let $\bar{\Lambda}_i = (\bar{\Lambda}_j - 1)$, the (21) is also satisfied at $\bar{Z}$.

We have verified the statement concerning the sequence $\{Z_k\}_{k=1}^{\infty}$ and the problem (15). Then directly get the statement that any accumulation point of $\{D_k, X_k, U_k, V_k\}_{k=1}^{\infty}$ is a KKT point of (7). Finally, from the equivalence between (1) and (7), the statement about $\{D_k, X_k\}_{k=1}^{\infty}$ and problem (1) can be verified. This completes the proof.

Far from being satisfactory, the above simple result nevertheless provides some assurance on the behavior of the ADMM-type algorithm applied to sparse dictionary learning problem. Further theoretical studies in this direction are certainly desirable.

4. Numerical tests

In this section, we conduct a synthetic test to verify the effectiveness of the proposed algorithm. We randomly generate a matrix $D \in \mathbb{R}^{40 \times 60}$ using the Matlab commend randn while each column of $D$ is normalized to unit 2-norm. Then construct a sparse matrix $X \in \mathbb{R}^{60 \times 1500}$ so that each column of $X$ has 3 non-zeros in random values (using randn) and at random locations. Then we synthesize the $40 \times 1500$ “exact” data matrix as the product $Y = DX$. 
We set the maximum number of iterations to \( \text{maxiter} = 1000 \) and the tolerance value to \( \text{tol} = 1 \times 10^{-06} \). The penalty parameters are set to \( \alpha = \|M\|_F / 100 \) and \( \beta = 0.1\alpha \) based on empirical evidence, since such a choice tends to give better convergence results.

We implement the proposed algorithm (9-14) to solve problem (1) with the above generating data \( \mathbf{Y} \). Figure 1 shows a representative convergence history of \( \|\mathbf{Y} - \mathbf{UV}\| \) and \( \|\mathbf{Y} - \mathbf{DX}\| \). From Figure 1, we can see that the data fidelity tends to \( 10^{-16} \), namely finding one of the optimal solutions. Besides, the solid and dashed line remain consistent after about 700 iterations, which implies \( D - U \to 0 \) and \( X - V \to 0 \). Therefore, it numerically demonstrates the result of convergence analysis.

![Figure 1. Convergence history of \( \|\mathbf{Y} - \mathbf{UV}\| \) and \( \|\mathbf{Y} - \mathbf{DX}\| \).](image)

5. Conclusions
Motivated by an alternating direction method and projection algorithm for structure-enforced matrix factorization, we extend the method to solve sparse dictionary learning and give a convergence result which is lack of research on theoretical analysis of multi-block and nonconvex ADMM. This provides some assurance on the behavior of the ADMM-type algorithm for sparse dictionary learning at least, although the simple result is far from being satisfactory. Further theoretical studies in this direction are certainly desirable.

Acknowledgments
This work has been supported by Natural Science Foundation of Liaoning Province of China (NO. 2019-BS-013) and in part by the Fundamental Research Funds for the Central Universities (NO. 3132020171).

References
[1] Mallat S and Zhang Z F 1993 Matching pursuit with time-frequency dictionaries IEEE Trans. Signal Process 41 pp 3397–3415
[2] Chen S, Billings S A and Luo W 1989 Orthogonal least squares methods and their application to non-linear system identification Int. J. Control 50 pp 1873–1896
[3] Tropp J A 2004 Greed is good: algorithmic results for sparse approximation IEEE Trans. Inform. Theory 50(10) pp 2231–2242
[4] Chen S, Donoho D and Saunders M 1998 Atomic decomposition by basis pursuit. SIAM J. Sci. Comput. 20 pp 33–61
[5] Gorodnitsky IF and Rao B D 1997 Sparse signal reconstruction from limited data using focuss: A re-weighted norm minimization algorithm IEEE Trans. Signal Process 45 pp 600–616
[6] Rao B D and Kuretz-Delgado K 1999 An affine scaling methodology for best basis selection IEEE Trans. Signal Process 47(1) pp 187–200
[7] Olshausen B A and Field D J 1996 Natural image statistics and efficient coding Network-Comput.
Neural 7(2) pp 333–339

[8] Lewicki M S and Sejnowski T J 2000 Learning overcomplete representations Neural Comput. 12 pp 337–365

[9] Engan K, Aase S O and Husoy J H 1999 Method of optimal directions for frame design IEEE Int. Conf. on Acoustics, Speech, and Signal Processing 5 pp 2443–2446

[10] Kreutz-Delgado K, Murray J F, Rao B D, Engan K, Lee T and Sejnowski T J 2003 Dictionary learning algorithms for sparse representation Neural Comput. 15(2) pp 349–396

[11] Aharon M, Elad M and Bruckstein A 2006 K-svd: An algorithm for designing overcomplete dictionaries for sparse representation IEEE Trans. Signal Process. 54(11) pp 4311–4322

[12] Mairal J, Bach F, Ponce J and Sapiro G 2009 Online dictionary learning for sparse coding The 26th Annual Int. Conf. on Machine Learning 8 pp 689–696

[13] Jenatton R, Audibert J Y and Bach F 2011 Structured variable selection with sparsity-inducing norms J. Mach. Learn. Res. 12 pp 2777–2824

[14] Szabo Z, Poczos B and Lorincz A 2011 Online group-structured dictionary learning 24th IEEE Conf. on Computer Vision and Pattern Recognition 11 pp 2865–2872

[15] Jenatton R, Obozinski G and Bach F 2010 Structured sparse principal component analysis Int. Conf. Artif. Intell. and Statistics (AISTATS)

[16] He J 1997 Semi-inverse method—a new approach to establishing variational principles for fluid mechanics. Journal Of Engineering Thermophysics 4

[17] He J 2000 Semi-Inverse Method and Generalized Variational Principles With Multi-Variables in Elasticity Applied Mathematics and Mechanics 21(7) pp 721–731

[18] He J and Wu X 2007 Variational iteration method: New development and applications, Computers & Mathematics With Applications 54(7-8) pp 881-894

[19] He J and Mo L 2013 Variational approach to the finned tube heat exchanger used in hydride hydrogen storage system International Journal Of Hydrogen Energy 38(36) pp 16177-16178

[20] Xu L, Yu B and Zhang Y 2017 An Alternating Direction and Projection Algorithm for Structure-enforced Matrix Factorization Comput. Optim. Appl. 68(2) pp 333–362

[21] Glowinski R and Marroco A 1975 Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de Dirichlet non linéaires. Revue Francaise d’Automatique, Informatique et Recherche Opérationnelle 9 pp 41–76

[22] Gabay D and Mercier B 1976 A dual algorithm for the solution of nonlinear variational problems via finite element approximation Comput. Math. Appl. 2(1) pp 17–40

[23] Zhang Y 2010 An alternating direction algorithm for nonnegative matrix factorization Technical Report TR10-03 CAAM Rice University

[24] Glowinski R and Le Tallec P 1989 Augmented Lagrangian and Operator Splitting Methods in Nonlinear Mechanics. SIAM, Philadelphia

[25] He B, Liao L, Han D and Yang H 2002 A new inexact alternating directions method for monotone variational inequalities. Math. Program. A(92) pp 103–118

[26] Deng W and Yin W 2012 On the global and linear convergence of the generalized alternating direction method of multipliers. Technical Report TR12-14 CAAM Rice University

[27] Chen C, He B, Ye Y, et al. 2016 The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent Math Program Ser A. 155(1-2) pp 57–79

[28] Li M, Sun D and Toh K-C 2015 A convergent 3-block semi-proximal ADMM for convex minimization problems with one strongly convex block. Asia-Pac J Oper. Res. 32(04)

[29] Cai X, Han D and Yuan X 2017 On the convergence of the direct extension of ADMM for three-block separable convex minimization models with one strongly convex function. Comput Optim Appl. 66(1) pp 39-73

[30] Chang X, Liu S and Zhao P 2018 Optimization 67(10) pp 1729-1743

[31] Guo K, Han D, Wang D and Wu T 2017 Front. Math. China 12(5) pp 1139-1162

[32] Wang Y, Yin W and Zeng J 2019 Journal of Scientific Computing 78(1) pp 29-63