The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Nonconvex Sparse Spectral Clustering by Alternating Direction Method of Multipliers and Its Convergence Analysis

Canyi Lu,[1] **Jiashi Feng,**[1] **Zhouchen Lin,**[2,3*] **Shuicheng Yan**[4,1]
[1] Department of Electrical and Computer Engineering, National University of Singapore
[2] Key Laboratory of Machine Perception (MOE), School of EECS, Peking University
[3] Cooperative Medianet Innovation Center, Shanghai Jiao Tong University
[4] 360 AI Institute
canyilu@gmail.com, elefjia@nus.edu.sg, zlin@pku.edu.cn, eleyans@nus.edu.sg

Abstract

Spectral Clustering (SC) is a widely used data clustering method which first learns a low-dimensional embedding U of data by computing the eigenvectors of the normalized Laplacian matrix, and then performs k-means on $U^\top$ to get the final clustering result. The Sparse Spectral Clustering (SSC) method extends SC with a sparse regularization on $UU^\top$ by using the block diagonal structure prior of $UU^\top$ in the ideal case. However, encouraging $UU^\top$ to be sparse leads to a heavily nonconvex problem which is challenging to solve and the work (Lu, Yan, and Lin 2016) proposes a convex relaxation in the pursuit of this aim indirectly. However, the convex relaxation generally leads to a loose approximation and the quality of the solution is not clear. This work instead considers to solve the nonconvex formulation of SSC which directly encourages $UU^\top$ to be sparse. We propose an efficient Alternating Direction Method of Multipliers (ADMM) to solve the nonconvex SSC and provide the convergence guarantee. In particular, we prove that the sequences generated by ADMM always exist a limit point and any limit point is a stationary point. Our analysis does not impose any assumptions on the iterates and thus is practical. Our proposed ADMM for nonconvex problems allows the stepsize to be increasing but upper bounded, and this makes it very efficient in practice. Experimental analysis on several real data sets verifies the effectiveness of our method.

Introduction

Data clustering is one of the most fundamental topics in unsupervised learning and has been widely applied in computer vision, data mining and many others. Clustering aims to divide the unlabeled data set into groups which consist of similar data points. Many clustering methods have been proposed up to now, e.g. k-means, spectral clustering (Ng et al. 2002; Shi and Malik 2000) and nonnegative matrix factorization (Lee and Seung 2001). The Spectral Clustering (SC) is one of the most widely used methods and it has a lot of applications in computer vision and signal analysis, e.g., image segmentation (Shi and Malik 2000), motion segmentation (Lu et al. 2012), and co-clustering problems of words and documents (Dhillon 2001).

*Corresponding author.

Assume that we are given n data points $X = [x_1, \cdots, x_n] = [X_1, \cdots, X_k] \in \mathbb{R}^{d\times n}$, where $X_j \in \mathbb{R}^{d\times n_j}$ denotes the j-th group with n_j points, $\sum_{j=1}^k n_j = n$ and k is the number of clusters. SC (Ng et al. 2002) partitions these n points into k clusters by the following procedures: First, compute an affinity matrix $W \in \mathbb{R}^{n\times n}$ with its element w_{ij} measuring the similarity between x_i and x_j. Second, construct the normalized Laplacian matrix $L = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, where D is a diagonal matrix with each diagonal element $d_{ii} = \sum_{j=1}^n w_{ij}$ and I is the identity matrix. Third, compute $U \in \mathbb{R}^{n\times k}$ by solving

$$\min_{U\in\mathbb{R}^{n\times k}} \langle UU^\top, L\rangle, \text{ s.t. } U^\top U = I. \tag{1}$$

Finally, compute $\hat{U} \in \mathbb{R}^{n\times k}$ by normalizing each row of U to have unit Euclidean length, treat the rows of $\hat{U}$ as data points in $\mathbb{R}^k$, and cluster them into k groups by k-means. Due to the significance of SC, many variants of SC have been proposed based on different ways of affinity matrix W construction and different normalizations of the Laplacian matrix L (Shi and Malik 2000; Von Luxburg 2007).

A recent work (Lu, Yan, and Lin 2016) proposes the Sparse Spectral Clustering (SSC) method which computes the low-dimensional embedding U in a different way:

$$\min_{U\in\mathbb{R}^{n\times k}} \left\langle L, UU^\top \right\rangle + \beta \left\| UU^\top \right\|_0, \text{ s.t. } U^\top U = I, \tag{2}$$

where $\beta > 0$ and $\|\cdot\|_0$ is the ℓ_0-norm which encourages $UU^\top$ to be sparse. The motivation for such a sparse regularizer is that $UU^\top$ is block diagonal (thus sparse) when W is block diagonal in the ideal case. Consider the ideal case that the affinity matrix W is block diagonal, i.e., $w_{ij} = 0$ if x_i and x_j are from different clusters. Let $C \in \mathbb{R}^{n\times k}$ denotes the indicator matrix whose row entries indicate to which group the points belong. That is, if x_i belongs to the group l, $c_{il} = 1$ and $c_{ij} = 0$ for all $j \neq l$. Then, for any orthogonal matrix $R \in \mathbb{R}^{k\times k}$, we have $\hat{U} = CR$. In this case, $\hat{U}\hat{U}^\top$ is block diagonal, i.e.,

$$\hat{U}\hat{U}^\top = CC^\top = \begin{bmatrix} \mathbf{1}_{n_1}\mathbf{1}_{n_1}^\top & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{n_2}\mathbf{1}_{n_2}^\top & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}_{n_k}\mathbf{1}_{n_k}^\top \end{bmatrix},$$

where $\mathbf{1}_m$ denotes the all one vector of length m and $\mathbf{0}$ is all one vector/matrix of proper size. Hence, $\hat{U}\hat{U}^\top$ implies the true membership of the data clusters and it is naturally sparse. Note that $\hat{U}$ is obtained by normalizing each row of U and thus $UU^\top$ is also sparse. However, such a block diagonal or sparse property may not appear in real applications since the affinity matrix W is usually not block diagonal. This motivates the sparse regularization on $UU^\top$ and thus leads to the SSC model in (2). However, the key challenge is that problem (2) is nonconvex and difficult to solve. The work (Lu, Yan, and Lin 2016) proposes a convex relaxation formulation as follows

$$\min_{P\in\mathbb{R}^{n\times n}} \langle P, L\rangle + \beta\|P\|_1, \text{ s.t. } \mathbf{0} \preceq P \preceq I, \text{ Tr}(P) = k, \quad (3)$$

where the ℓ_1-norm $\|\cdot\|_1$ is used as a surrogate of ℓ_0-norm while the nonconvex constraint consisting of all the fixed rank projection matrices, i.e., $\{UU^\top | U^\top U = I\}$, is replaced as its convex hull $\{P \in \mathbb{S}^{n\times n} | 0 \preceq P \preceq I, \text{Tr}(P) = k\}$ (Fillmore and Williams 1971). Here, $\mathbb{S}^n$ denotes the set of symmetric matrices. For $A, B \in \mathbb{S}^n$, $A \preceq B$ means that $B - A$ is positive semi-definite. Problem (3) is convex and the optimal solution can be computed by Alternating Direction Method of Multipliers (ADMM) (Gabay and Mercier 1976; Lu et al. 2017), which is efficient in practice. After solving (3) with the solution P^*, the low-dimensional embedding U of data X can be approximated by using the first k eigenvectors corresponding to the largest k eigenvalues of P. This is equivalent to computing U by solving

$$\min_{U\in\mathbb{R}^{n\times k}} \left\|P^* - UU^\top\right\|, \text{ s.t. } U^\top U = I, \quad (4)$$

where $\|\cdot\|$ denotes the Frobenius norm of a matrix. After obtaining U, one is able to cluster the data points into k groups as that in SC.

From the above discussions, it can be seen that a main limitation of the convex SSC relaxation (3) is that the obtained solution may be far from optimal to (2). The reason is that the difference $\left\|P^* - UU^\top\right\|$ in (4) is not guaranteed to be 0 or sufficiently small. Thus, enforcing P to be sparse in (3) does not guarantee a sparse $UU^\top$ in (4). It is obvious that such an issue is mainly caused by the usage of the relaxation of the convex hull $\{P \in \mathbb{S}^{n\times n} | 0 \preceq P \preceq I, \text{Tr}(P) = k\}$ instead of $\{UU^\top | U^\top U = I\}$.

In this work, we aim to address the above possibly loose relaxation issue of the convex SSC model by directly solving the following nonconvex problem

$$\min_{U\in\mathbb{R}^{n\times k}} \left\langle L, UU^\top\right\rangle + g(UU^\top), \text{ s.t. } U^\top U = I, \quad (5)$$

where $g : \mathbb{R}^{n\times n} \to \mathbb{R}$ is a sparse regularizer. The choice of the spare regularizer is not very important in SSC. We allow g to be nonconvex (see more detailed assumption in the next section). Problem (5) is nonconvex and challenging to solve due to the orthogonal constraint and the complex term $g(UU^\top)$. This work proposes to solve it by Alternating Direction Method of Multipliers (ADMM) and provide the convergence guarantee. Our proposed ADMM is flexible as we allow the stepsize in ADMM to be monotonically increasing (but upper bounded). Such a choice of stepsize is widely used and has been verified to be effective in improving the convergence of ADMM for convex optimization. Note that we are the first one to use such a choice of stepsize in ADMM for nonconvex problems and provide its support in theory. When characterizing the convergence of ADMM, we show that the augmented Lagrangian function value is monotonically decreasing. This further guarantees that the sequences generated by the proposed ADMM are bounded and there always exists a limit point and any limit point is a stationary point.

Algorithm 1 Solve (6) by ADMM

Initialize: $\rho > 1$, $\mu_{\max}$, $k = 0$, P_k, U_k, Y_k, μ_k.
while not converged **do**

1. Compute U_{k+1} by solving (8);
2. Compute P_{k+1} by solving (9);
3. Compute Y_{k+1} by (10);
4. Compute μ_{k+1} by (11);
5. $k = k + 1$.

end while

The Proposed ADMM Algorithm

In this section, we present the ADMM algorithm for solving the nonconvex problem (5). We first reformulate it as the following equivalent problem

$$\begin{aligned} &\min_{P\in\mathbb{R}^{n\times n}, U\in\mathbb{R}^{n\times k}} \left\langle L, UU^\top\right\rangle + g(P), \\ &\text{s.t. } P = UU^\top, \ U^\top U = I. \end{aligned} \quad (6)$$

The standard augmented Lagrangian function is

$$\begin{aligned} \mathcal{L}(P, U, Y_1, Y_2, \mu) = &\left\langle L, UU^\top\right\rangle + g(P) + \left\langle Y_1, P - UU^\top\right\rangle \\ &+ \left\langle Y_2, UU^\top - I\right\rangle + \frac{\mu}{2}\left\|P - UU^\top\right\|^2 + \frac{\mu}{2}\left\|U^\top U - I\right\|^2, \end{aligned}$$

where Y_1 and Y_2 are the dual variables and $\mu > 0$. However, it is difficult to update U by minimizing the above augmented Lagrangian function when fixing other variables. To update U efficiently, we instead use the following *partial* augmented Lagrangian function

$$\begin{aligned} \mathcal{L}(P, U, Y, \mu) = &\left\langle L, UU^\top\right\rangle + g(P) + \left\langle Y, P - UU^\top\right\rangle \\ &+ \frac{\mu}{2}\left\|P - UU^\top\right\|^2. \end{aligned} \quad (7)$$

Then we can solve problem (6) by Alternating Direction Method of Multipliers by the following rules.
1. Fix $P = P_k$ and update U by

$$\begin{aligned} U_{k+1} &= \operatorname*{argmin}_{U\in\mathbb{R}^{n\times k}} \mathcal{L}(P_k, U, Y_k, \mu_k), \text{ s.t. } U^\top U = I. \\ &= \operatorname*{argmin}_{U} \left\|UU^\top - P_k + (L - Y_k)/\mu_k\right\|^2, \\ &\quad \text{s.t. } U^\top U = I. \end{aligned} \quad (8)$$

2. Fix $U = U_{k+1}$ and update P by

$$\begin{aligned} P_{k+1} &= \underset{P}{\operatorname{argmin}}\ \mathcal{L}(P, U_{k+1}, Y_k, \mu_k) \\ &= \underset{P}{\operatorname{argmin}}\ g(P) + \frac{\mu_k}{2}\left\|P - U_{k+1}U_{k+1}^\top + Y_k/\mu_k\right\|^2. \end{aligned} \tag{9}$$

3. Update the dual variable by

$$Y_{k+1} = Y_k + \mu_k(P_{k+1} - U_{k+1}U_{k+1}^\top). \tag{10}$$

4. Update the stepsize μ by

$$\mu_{k+1} = \min(\mu_{\max}, \rho\mu_k),\ \rho > 1. \tag{11}$$

The whole procedure of ADMM for (6) is given in Algorithm 1. It can be seen that the U-subproblem (8) has a closed form solution. The P-subproblem (9) requires computing the proximal mapping of g. It usually has a closed form solution when g is simple.

We would like to emphasize that, for nonconvex optimization, our ADMM allows the stepsize μ_k to be increasing (but upper bounded), while previous nonconvex ADMM methods have to fix it. Such a choice of stepsize has been verified to be effective in improving the convergence of ADMM for convex optimization in practice and the convergence guarantee is also known (Lu et al. 2017; Lin, Chen, and Ma 2010). To the best of our knowledge, this is the first work which uses varying stepsize in ADMM for nonconvex problems and the convergence analysis for supporting this is also different from the convex case.

Main Result: The Convergence Analysis

The most important contribution is the convergence analysis of the proposed ADMM in Algorithm 1 for nonconvex problems which is generally challenging. This section gives the details of the convergence results. We first introduce the subgradient of any function (Rockafellar and Wets 2009), which will be used later.

Definition 1. *Let $S \subseteq \mathbb{R}^m$ and $x_0 \in S$. A vector v is normal to S at x_0 in the regular sense, denoted as $v \in \hat{N}_S(x_0)$, if*

$$\langle v, x - x_0\rangle \le o(\|x - x_0\|),\ x \in S,$$

where $o(\|y\|)$ is defined by $\lim_{\|y\|\to 0} \frac{o(\|y\|)}{\|y\|} = 0$. A vector is normal to S at x_0 in the general sense, denoted as $v \in N_S(x_0)$, if there exist sequences $\{x^k\} \subset S$, $\{v^k\}$ such that $x^k \to x_0$ and $v^k \to v$ with $v^k \in \hat{N}_S(x^k)$. The cone $N_S(x_0)$ is called the normal cone to S at x_0.

Definition 2. *Consider a lower semi-continuous function $h : \mathbb{R}^m \to (-\infty, +\infty]$ and a point x_0 with $h(x_0)$ finite. For a vector $v \in \mathbb{R}^m$, one says that*

(a) v is a regular subgradient of h at x_0, denoted as $v \in \hat{\partial} h(x_0)$, if

$$h(x) \ge h(x_0) + \langle v, x - x_0\rangle + o(\|x - x_0\|);$$

(b) v is a (general) subgradient of h at x_0, denoted as $v \in \partial h(x_0)$, if there exist sequences $\{x^k\}$, $\{v^k\}$ such that $x^k \to x_0$, $h(x_k) \to h(x_0)$ and $v^k \in \hat{\partial} h(x^k)$ with $v^k \to v$.

Let S be a closed non-empty subset of $\mathbb{R}^m$ and its indicator function be

$$\iota_S(x) = \begin{cases} 0, & \text{if } x \in S, \\ +\infty, & \text{otherwise.} \end{cases}$$

Then its subgradient is $\partial \iota_S(x_0) = N_S(x_0), x_0 \in S$. In this work, we denote $\mathcal{O} = \{U \in \mathbb{R}^{n\times k} | U^\top U = I\}$ and the indicator function as $\iota_{\mathcal{O}}(U)$.

To guarantee the convergence, we need some assumptions for problem (6) as follows:

A1. L is positive semi-definite.

A2. $g : \mathbb{R}^{n\times n} \to R$ is lower bounded, differential and ∇g is Lipschitz continuous, i.e., there exists $l > 0$ such that

$$\|\nabla g(X) - \nabla g(Y)\| \le l\|X - Y\|,\ \forall X, Y \in \mathbb{R}^{n\times n}.$$

A3. The stepsize μ_k is chosen large enough such that

(1) The P-subproblem (9) is strongly convex with modulus γ_k.

(2) $\mu_k\gamma_k > l^2(\rho + 1)$ and $\mu_k \ge l$.

We have some remarks regarding the above assumptions. First, A1 holds for the SSC model since L is the normalized Laplacian matrix; Second, g can be nonconvex. In SSC, one may use convex or nonconvex sparse regularizer g. But g should be Lipschitz differential which can be achieved by using the smoothing technique (Nesterov 2005) if necessary (see an example in the Experiment section); Third, the P-subproblem (9) is eventually strongly convex when μ_k is large enough.

At the following, we will prove several properties of Algorithm 1 and give the convergence results.

Lemma 1. *Under assumptions A1-A3, all the subproblems in Algorithm 1 are well defined.*

Proof. The P-subproblem (9) is well defined since g is lower bounded under assumption A2. Also, it is obvious that the U-subproblem (8) is well defined. □

Lemma 2. *Under assumptions A1-A3, we have*

$$\|Y_k - Y_{k+1}\|^2 \le l^2\|P_k - P_{k+1}\|^2. \tag{12}$$

Proof. From the P-subproblem (9), we have the following optimality condition

$$\nabla g(P_{k+1}) + Y_k + \mu_k(P_{k+1} - U_{k+1}U_{k+1}^\top) = 0. \tag{13}$$

By using $Y_{k+1} = Y_k + \mu_k(P_{k+1} - U_{k+1}U_{k+1}^\top)$ in (10), we have

$$\nabla g(P_{k+1}) = -Y_{k+1}. \tag{14}$$

Then we have

$$\|Y_{k+1} - Y_k\| = \|\nabla g(P_{k+1}) - \nabla g(P_k)\| \le l\|P_{k+1} - P_k\|,$$

where the last inequality uses assumption A2. The proof is completed. □

Lemma 3. *Under assumptions A1-A3, the sequences $\{P_k, U_k, Y_k\}$ generated by Algorithm 1 satisfy*

(a) $\mathcal{L}(P_k, U_k, Y_k, \mu_k)$ *is monotonically decreasing, i.e.,*

$$\begin{aligned}&\mathcal{L}(P_{k+1}, U_{k+1}, Y_{k+1}, \mu_{k+1}) - \mathcal{L}(P_k, U_k, Y_k, \mu_k)\\ \leq &-\left(\frac{\gamma_k}{2} - \frac{l^2(\rho+1)}{2\mu_k}\right)\|P_{k+1} - P_k\|^2. \end{aligned} \tag{15}$$

(b) $\lim_{k\to+\infty} \mathcal{L}(P_k, U_k, Y_k, \mu_k) = \mathcal{L}^*$ *for some constant* $\mathcal{L}^*$.

(c) When $k \to +\infty$, $P_{k+1} - P_k \to 0$, $Y_{k+1} - Y_k \to 0$ *and* $P_k - U_k U_k^\top \to 0$.

(d) The sequences $\{P_k\}$, $\{U_k\}$ *and* $\{Y_k\}$ *are bounded.*

(e) There exists $G = [G_P\ G_U\ G_Y]$, *where*

$$\begin{aligned}G_P =&\partial_P \mathcal{L}(P_{k+1}, U_{k+1}, Y_{k+1}, \mu_k),\\ G_U \in&\partial_U \mathcal{L}(P_{k+1}, U_{k+1}, Y_{k+1}, \mu_k) + \partial_U \iota_{\mathcal{O}}(U_{k+1}),\\ G_Y =&\partial_Y \mathcal{L}(P_{k+1}, U_{k+1}, Y_{k+1}, \mu_k),\end{aligned}$$

such that

$$\begin{aligned}\|G\|^2 \leq&(8d + 1 + \frac{1}{\mu_0^2})\|Y_k - Y_{k+1}\|^2\\ &+ 8d\mu_{\max}^2\|P_k - P_{k+1}\|^2. \end{aligned} \tag{16}$$

The proof of Lemma 3 can be found in the supplementary material. The property (12) is important for proving (15), which guarantees that $\mathcal{L}(P_k, U_k, Y_k, \mu_k)$ is monotonically decreasing due to the choice of μ_k in assumption A3. This combined with the lower bounded property guarantees that $\mathcal{L}(P_k, U_k, Y_k, \mu_k)$ converges to some $\mathcal{L}^* > 0$. For convex problems, there have several different quantities to characterize the convergence of ADMM, see (Lu et al. 2017; He and Yuan 2012; Liu, Lin, and Su 2013). However, they are not applicable to ADMM for nonconvex problems. Here, the convergence is characterizing based on a different way by using the decreasing sequence $\mathcal{L}(P_k, U_k, Y_k, \mu_k)$. Note that the decreasing property of $\mathcal{L}(P_k, U_k, Y_k, \mu_k)$ does not necessary hold for ADMM for convex optimization. This difference implies that the nonconvex problems which can be solved by ADMM are relatively limited and the convergence guarantee of ADMM for nonconvex problems is much more challenging. Based on (15), many other properties are proved. For example, Lemma 3 (c) gives some necessary results when the algorithm converges and (d) and (e) are important for proving the convergence to stationary point shown below.

Theorem 1. *Assume that the assumptions A1-A3 are satisfied. Let* (P^*, U^*, Y^*) *denotes any limit point of the sequence* $\{P_k, U_k, Y_k\}$ *generated by Algorithm 1. Then the limit point is a stationary point of problem (6), i.e.,*

$$0 \in \partial_U \mathcal{L}(P^*, U^*, Y^*, \mu^*) + \partial_U \iota_{\mathcal{O}}(U^*), \tag{17}$$

$$0 = \partial_P \mathcal{L}(P^*, U^*, Y^*, \mu^*), \tag{18}$$

$$0 = \partial_Y \mathcal{L}(P^*, U^*, Y^*, \mu^*) = P^* - U^* {U^*}^\top. \tag{19}$$

Proof. From the boundedness of $\{P_k, U_k, Y_k, \mu_k\}$ in Lemma 3, there exists a convergent subsequence and a limit point, denoted by $(P_{k_j}, U_{k_j}, Y_{k_j}, \mu_{k_j}) \to (P^*, U^*, Y^*, \mu^*)$ as $j \to +\infty$. Then, by using $P_{k+1} - P_k \to 0$, $Y_{k+1} - Y_k \to 0$ and (16) in Lemma 3, for $k \geq 1$, there exists $G_k \in \partial\mathcal{L}(P_k, U_k, Y_k, \mu_{k-1})$ such that $\|G_k\| \to 0$. In particular, $\|G_{k_j}\| \to 0$ as $j \to +\infty$. By the definition of general subgradient, we have $0 \in \partial\mathcal{L}(P^*, U^*, Y^*, \mu^*)$. This implies that (17)-(19) hold. Thus, any limit point is a stationary point. □

Theorem 2. *For every* $K \geq 1$, *the sequences* $\{P_k, U_k, Y_k\}$ *generated by Algorithm 1 satisfies*

$$\min_{0\leq k\leq K}\|P_{k+1} - P_k\|^2 \leq \frac{\mathcal{L}(P_0, U_0, Y_0, \mu_0) - \mathcal{L}^*}{(K+1)c_K},$$

$$\min_{0\leq k\leq K}\|Y_{k+1} - Y_k\|^2 \leq \frac{l^2(\mathcal{L}(P_0, U_0, Y_0, \mu_0) - \mathcal{L}^*)}{(K+1)c_K},$$

$$\min_{0\leq k\leq K}\left\|P_{k+1} - U_{k+1}U_{k+1}^\top\right\|^2 \leq \frac{l^2(\mathcal{L}(P_0, U_0, Y_0, \mu_0) - \mathcal{L}^*)}{(K+1)c_K\mu_0^2},$$

where $c_K = \min_{0\leq k\leq K}\left(\frac{\gamma_k}{2} - \frac{l^2(\rho+1)}{2\mu_k}\right) > 0$.

Proof. From (15) and the definition of $c_K > 0$, we have

$$\begin{aligned}&c_K\|P_{k+1} - P_k\|^2\\ \leq&\mathcal{L}(P_k, U_k, Y_k, \mu_k) - \mathcal{L}(P_{k+1}, U_{k+1}, Y_{k+1}, \mu_{k+1}).\end{aligned}$$

Summing the above equality over $k = 0, \cdots, K$, we obtain

$$\begin{aligned}&\sum_{k=0}^{K} c_K\|P_{k+1} - P_k\|^2\\ \leq&\mathcal{L}(P_0, U_0, Y_0, \mu_0) - \mathcal{L}(P_{k+1}, U_{k+1}, Y_{k+1}, \mu_{k+1})\\ \leq&\mathcal{L}(P_0, U_0, Y_0, \mu_0) - \mathcal{L}^*.\end{aligned}$$

Thus, we have

$$\begin{aligned}&\min_{0\leq k\leq K}\|P_{k+1} - P_k\|^2\\ \leq&\frac{1}{K+1}\sum_{k=0}^{K}\|P_{k+1} - P_k\|^2 \leq \frac{\mathcal{L}(P_0, U_0, Y_0, \mu_0) - \mathcal{L}^*}{(K+1)c_K}.\end{aligned}$$

The proof is completed by further using (12) and (10). □

Theorem 2 gives the $O(1/K)$ convergence rate of our proposed ADMM based on the smallest difference of iterates of P_k, Y_k and the residual. To the best of our knowledge, this is the first convergence rate of ADMM for nonconvex problems. In theory, such a result is relatively weaker than the convex case since the used measure is $\min_{0\leq k\leq K}\|P_{k+1} - P_k\|^2$ but not $\|P_{K+1} - P_K\|^2$. But in practice, we observe that the sequence $\|P_{k+1} - P_k\|^2$ seems to decrease (see Figure 2 (b) in the Experiment section), though this is currently not clear in theory. This observation in practice implies that the above convergence rate makes sense.

It is worth mentioning that the convergence guarantee of ADMM for convex problems has been well established (Lu et al. 2017; He and Yuan 2012; Liu, Lin, and Su 2013). However, for nonconvex cases, the convergence analysis of ADMM for different nonconvex problems is quite different. There are some recent works (Hong, Luo, and Razaviyayn 2016; Wang, Yin, and Zeng 2015) which apply ADMM to

solve nonconvex problems and provide some analysis. However, these works are not able to solve our problem (6) since the constraints in their considered problems should be relatively simple while our problem has a special nonconvex constraint $P = UU^\top$. The work (Hong, Luo, and Razaviyayn 2016) is not applicable to our problem since it requires all the subproblems to be strongly convex while our U-subproblem (8) is nonconvex. When considering to apply the method in (Wang, Yin, and Zeng 2015) to solve (5), one has to exactly solve the problem of the following type

$$\min_P \; g(PP^\top) + \frac{1}{2}\|P - B\|^2,$$

in each iteration. This is generally very chellenging even when g is convex. Also, we would like to emphasize that our ADMM allows the stepsize μ_k to be increasing (but upper bounded), while previous nonconvex ADMM algorithms simply fix it. Though our considered problem is specific, the analysis for the varying stepsize μ_k is applicable to other nonconvex problems in (Hong, Luo, and Razaviyayn 2016). In practice, the convergence speed of ADMM is sensitive to the choice of μ, but it is generally difficult to find a proper constant stepsize for fast convergence. Our choice of μ_k has been shown to be effective in improving the convergence speed and widely used in convex optimization (Lu et al. 2017; Lin, Chen, and Ma 2010). In practice, we find that such a technique is also very useful for fast implementation of ADMM for nonconvex problems. We are also the first one to give the convergence rate (in the sense of Theorem 2) of ADMM for nonconvex problems.

Experiments

In this section, we conduct some experiments to analyze the convergence of the proposed ADMM for nonconvex SSC and show its effectiveness for data clustering. We consider to solve the following nonconvex SSC model

$$\begin{aligned} &\min_{P\in\mathbb{R}^{n\times n}, U\in\mathbb{R}^{n\times k}} \; \langle L, UU^\top\rangle + g_\sigma(P), \\ &\text{s.t. } P = UU^\top,\; U^\top U = I, \end{aligned} \tag{20}$$

where g_σ is the smoothed ℓ_1-norm $\beta\|P\|_1$ with a smoothness parameter $\sigma > 0$ defined as follows

$$g_\sigma(P) = \max_Z \langle P, Z\rangle - \frac{\sigma}{2}\|Z\|^2, \text{ s.t. } \|Z\|_\infty \le \beta, \tag{21}$$

where $\|Z\|_\infty = \max_{ij}|z_{ij}|$. According to Theorem 1 in (Nesterov 2005), the gradient of $g_\sigma(P)$ is given by $\nabla g_\sigma(P) = \min\{\beta, \max\{P/\sigma, -\beta\}\}$ and is Lipschitz continuous with Lipschitz constant $l = 1/\sigma$. Note that g_σ is convex. So we set $\mu_0 = 1.01(l\sqrt{\rho+1})$, which guarantees the assumption A3 holds. In Algorithm 1, we set ρ= 1.05, $\mu_{\max} = 1e10$, and U_0 is initialized as the k eigenvectors associated to the k smallest eigenvalues of L, where k is the number of the clusters and L is the normalized Laplacian matrix constructed based on the given affinity matrix W. Then we set $P_0 = U_0U_0^\top$ and $Y_0 = 0$. We use the following stopping criteria for Algorithm 1

$$\max\{\|P_{k+1} - P_k\|_\infty, \|P_{k+1} - U_{k+1}U_{k+1}^\top\|_\infty\} \le 10^{-6}, \tag{22}$$

Table 1: Clustering errors (%) on the Extended Yale B database based on the *sparse* affinity matrix W constructed by the ℓ_1-graph.

# of subjects	SC	SSC	SSC-PG	SSC-ADMM
2	1.56±2.95	1.80±2.89	1.37±3.15	**1.21±2.10**
3	3.26±7.69	3.36±7.76	3.12±6.23	**2.40±4.92**
5	6.33±5.36	6.61±5.93	5.65±4.33	**3.86±2.82**
8	8.93±6.11	4.98±4.00	4.95±3.36	**4.67±3.40**
10	9.94±4.57	**4.60±2.59**	5.91±4.52	5.84±3.43

which is implied by our convergence analysis. For all the experiments, we use $\sigma = 0.01$ (in practice, we find that the clustering performance is not sensitive when $\sigma \le 0.01$).

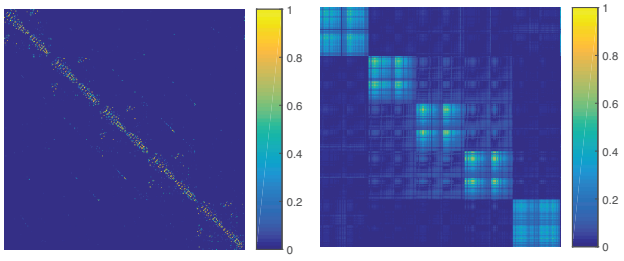
We conduct two experiments based on two different ways of affinity matrix construction. The first experiment considers the subspace clustering problem in (Elhamifar and Vidal 2013). A *sparse* affinity matrix W is computed by using the sparse subspace clustering method (ℓ_1-graph) (Elhamifar and Vidal 2013), and then it is used as the input for SC, convex SSC (Lu, Yan, and Lin 2016) and our nonconvex SSC model solved by ADMM. The second experiment instead uses the Gaussian kernel to construct the affinity matrix which is generally *dense*. We will show the effectiveness of nonconvex SSC in both settings.

Affinity matrix construction by the ℓ_1-graph

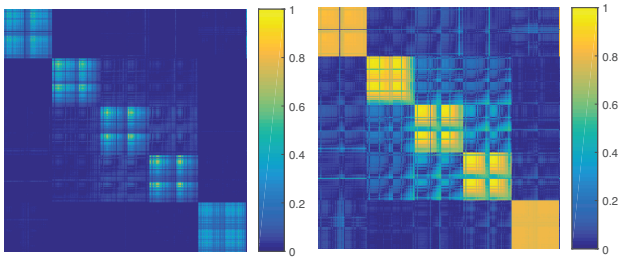
For the first experiment, we consider the case that the affinity matrix is constructed by the ℓ_1-graph (Elhamifar and Vidal 2013). We test on the Extended Yale B database (Georghiades, Belhumeur, and Kriegman 2001) to analyze the effectiveness of our nonconvex SSC model in (20). The Extended Yale B dataset consists of 2,414 face images of 38 subjects. Each subject has 64 faces. We resize the images to 32×32 and vectorized them as 1,024-dimensional data points. We construct 5 subsets which consist of all the images of the randomly selected 2, 3, 5, 8 and 10 subjects of this dataset. For each trial, we follow the settings in (Elhamifar and Vidal 2013) to construct the affinity matrix W by solving a sparse representation model (or ℓ_1-graph), which is the most widely used method. Based on the learned affinity matrix by ℓ_1-graph, the following three methods are compared to find the final clustering results:

- SC: traditional spectral clustering method (Ng et al. 2002).
- SSC: convex SSC model (Lu, Yan, and Lin 2016).
- SSC-ADMM: our nonconvex SSC model solved by the proposed ADMM in Algorithm 1.
- SSC-PG: our nonconvex SSC model (5) can also be solved by the Proximal Gradient (PG) (Beck and Teboulle 2009) method (a special case of Algorithm 1 in (Mairal 2013)). In each iteration, PG updates U_{k+1} by

$$\begin{aligned} U_{k+1} = \; &\operatorname*{argmin}_{U\in\mathbb{R}^{n\times k}} g(U_kU_k^\top) + \left\langle \nabla g(U_kU_k^\top), UU^\top - U_kU_k^\top\right\rangle \\ &+ \frac{l}{2}\left\|UU^\top - U_kU_k^\top\right\|^2 + \left\langle L, UU^\top\right\rangle \\ &\text{s.t. } U^\top U = I. \end{aligned}$$



(a) W learned by the ℓ_1-graph (b) $UU^\top$ by SC



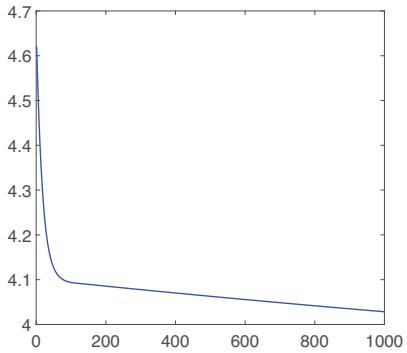
(c) $UU^\top$ by convex SSC (d) $UU^\top$ by SSC-ADMM

Figure 1: An example with 5 subjects from the Extended Yale B database. (a) Plot of the affinity matrix W learned by the ℓ_1-graph (Elhamifar and Vidal 2013); (b) Plot of $UU^\top$ with U learned by SC in (1); (c) Plot of $UU^\top$ with U learned by convex SSC in (4); (d) Plot of $UU^\top$ with U learned by SSC-ADMM in (6). Each matrix is normalized to [0,1] for better visualization.

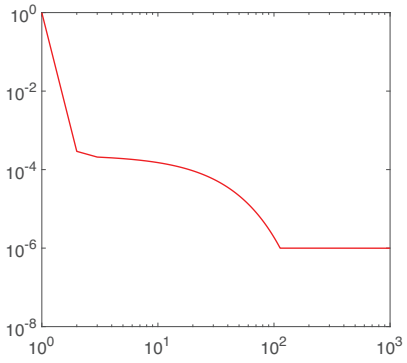
It is easy to see that the above problem has a closed form solution. We use the stopping criteria $\|U_k - U_{k+1}\|_\infty \leq 10^{-6}$. We name the above method as SSC-PG.

Note that all the above four methods use the same affinity matrix as the input and their main difference is the different ways of learning of low-dimensional embedding U. In the nonconvex model (20), we set $\beta = 0.01$. The experiments are repeated 20 times and the mean and standard deviation of the clustering error rates (see the definition in (Elhamifar and Vidal 2013)) are reported.

The clustering results are shown in Table 1. It can be seen that our nonconvex SSC models outperform convex SSC in most cases. The main reason is that nonconvex SSC is able to directly encourage $UU^\top$ to be sparse while SSC achieves this in a two-stage way (required solving (3) and (4)). Considering a clustering example with $k = 5$ subjects from the Yale B dataset, Figure 1 plots the learned affinity matrix W by ℓ_1-graph, and $UU^\top$ learned by SC, SSC and SSC-ADMM, respectively. Note that $UU^\top$ is important for data clustering since $\hat{U}\hat{U}^\top$ ($\hat{U}$ is the row normalization of U) implies the true membership of the data clusters in the ideal case (see the discussions in the Introduction section). It can be seen that $UU^\top$ by SSC-ADMM looks more discriminative since the within-cluster connections are much stronger than the between-cluster connections. Also, for the convergence of the proposed ADMM, we plot the augmented Lagrangian function $\mathcal{L}(P_k, U_k, Y_k, \mu_k)$ and the stopping criteria in (22). It can be seen that $\mathcal{L}$ is monotonically decreasing



(a) $\mathcal{L}(P_k, U_k, Y_k, \mu_k)$



(b) stopping criteria

Figure 2: Plots of (a) convergence of $\mathcal{L}(P_k, U_k, Y_k, \mu_k)$ v.s. k and (b) convergence of the stopping criteria in (22) v.s. k. The used data is a 5 subjects subset of Extended Yale B database.

and the stopping criteria converges towards 0. The convergence behavior is consistent with our theoretical analysis.

Affinity matrix construction by the Gaussian kernel

For the second experiment, we consider the case that the affinity matrix is constructed by the Gaussian kernel. We test on 10 datasets which are of different sizes and are widely used in pattern analysis. They include 5 datasets (Wine, USPS, Glass, Letter, Vehicle) from the UCI website (Lichman 2013)[1], UMIST[2], PIE (Sim, Baker, and Bsat 2002), COIL20[3], CSTR[4] and AR (Martinez 1998). For some datasets, e.g., USPS and Letter, we use a subset instead due to the relatively large size. For some image datasets, e.g., UMIST, PIE, and COIL20, the images are resized to 32×32 and then vectorized as features. The statistics of these datasets are summarized in Table 2. The key difference of this experiment from the first one is that we use the Gaussian kernel instead of the sparse subspace clustering to

[1]https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

[2]https://www.sheffield.ac.uk/eee/research/iel/research/face

[3]http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php

[4]http://www.cs.rochester.edu/trs/

Table 2: Statistics of 10 datasets.

dataset	# samples	# features	# clusters
Wine	178	13	3
USPS	1,000	256	10
Glass	214	9	6
Letter	1,300	16	26
Vehicle	846	18	4
UMIST	564	1,024	20
PIE	1,428	1,024	68
COIL20	1,440	1,024	20
CSTR	476	1,000	4
AR	840	768	120

Table 3: Clustering accuracy on 10 datasets. In this table, SC, SSC, SSC-PG and SSC-ADMM use the *dense* affinity matrix W constructed by the Gaussian kernel.

	k-means	NMF	SC	SSC	SSC-PG	SSC-ADMM
Wine	94.4	96.1	94.9	96.1	96.6	**97.2**
USPS	67.3	69.2	71.2	73.4	**76.8**	**76.8**
Glass	40.7	39.3	41.1	43.0	44.9	**45.3**
Letter	27.1	30.4	31.8	35.3	**36.4**	36.0
Vehicle	62.1	61.3	67.0	70.0	73.0	**73.4**
UMIST	52.1	63.8	63.3	64.2	65.1	**66.1**
PIE	35.4	37.9	42.0	46.7	46.8	**51.1**
COIL20	59.0	46.2	63.1	64.5	**69.1**	67.9
CSTR	65.0	70.0	68.9	72.7	71.0	**76.3**
AR	24.2	35.0	36.1	37.1	37.7	**39.0**

construct the affinity matrix W.

The Gaussian kernel parameter is tuned by the grid $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$. We use the same affinity matrix W as the input of SC, SSC (Lu, Yan, and Lin 2016), SSC-PG and SSC-ADMM. The parameter β in SSC, SSC-PG and SSC-ADMM is searched from $\{10^{-4}, 10^{-3}, 10^{-2}\}$. We further compare the these methods with k-means and Nonnegative Matrix Factorization (NMF). Under each parameter setting of each method mentioned above, we repeat the clustering for 20 times, and compute the average result. We report the best average accuracy for each method in Table 3.

From Table 3, we have the following observations. First, it can be seen that the SSC models (SSC, SSC-PG and SSC-ADMM) improve the traditional SC and our SSC-ADMM achieves the best performances in most cases. Second, this experiment not only verifies the superiority of SSC over SC, but also shows the importance of the nonconvex SSC and the effectiveness of our solver. Third, this experiment implies that the nonconvex SSC improves the convex SSC when given the dense affinity matrix constructed by the Gaussian kernel which is different from the sparse ℓ_1-graph in the first experiment. Beyond the accuracy, we further compare the efficiency of SSC-PG and SSC-ADMM which use different solvers for the equivalent nonconvex SSC model. Figure 3 gives the average running time of both methods. It can be seen that SSC-ADMM is much more efficient than SSC-PG. The main reason behind is that SSC-PG has to construct a relatively loose majorant surrogate for g in each iteration (Mairal 2013) and thus requires many more (usually more than $1,000$) iterations. Note that the same phenomenon appears in the convex optimization (Lin, Chen, and Ma 2010).

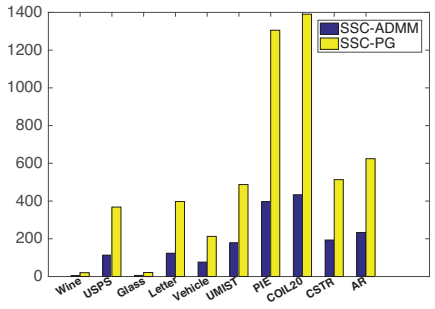


Figure 3: Comparison of the average running time (in second) of SSC-ADMM and SSC-PG on 10 datasets.

Conclusion and Future Work

This paper addressed the loose convex relaxation issue of SSC proposed in (Lu, Yan, and Lin 2016). We proposed to use ADMM to solve the nonconvex SSC problem (6) directly instead of the convex relaxation. More importantly, we provided the convergence guarantee of ADMM for such a nonconvex problem which is challenging and has not been addressed before. It is worth mentioning that our ADMM method and analysis allow the stepsize to be monotonically increased (but upper bounded). Such a technique has been verified to be effective in improving the convergence in practice for convex optimization and this is the first work which introduces it to ADMM for nonconvex optimization. Also, our convergence guarantee generally requires relatively weak assumptions, e.g., no assumption on the iterates and the subproblems are not necessarily to be strongly convex. Thus it is more practical and can be used to solve other related problems. Beyond the convergence guarantee, we also gave some experimental studies to verify our analysis and the clustering results on several real datasets demonstrated the improvement of nonconvex SSC over convex SSC.

There have some interesting future works. Though the solved nonconvex problem in this work is specific, the problems with nonconvex orthogonal constraint are interesting and such a nonconvex constraint or related ones appear in many models in component analysis. It will be interesting to apply ADMM to solve other problems with similar constraints and provide the convergence guarantee. It will be also interesting to apply our technique to solve some other nonconvex low rank minimization models in (Lu et al. 2016; 2015).

Acknowledgements

J. Feng is partially supported by National University of Singapore startup grant R-263-000-C08-133 and Ministry of Education of Singapore AcRF Tier One grant R-263-000-C21-112. Z. Lin is supported by National Basic Research Program of China (973 Program) (Grant no. 2015CB352502) and National Natural Science Foundation

(NSF) of China (Grant nos. 61625301 and 61731018).

References

Beck, A., and Teboulle, M. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*.

Dhillon, I. S. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*, 269–274. ACM.

Elhamifar, E., and Vidal, R. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *TPAMI* 35(11):2765–2781.

Fillmore, P., and Williams, J. 1971. Some convexity theorems for matrices. *Glasgow Mathematical Journal* 12(02):110–117.

Gabay, D., and Mercier, B. 1976. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications* 2(1):17–40.

Georghiades, A. S.; Belhumeur, P. N.; and Kriegman, D. J. 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. *TPAMI* 23(6):643–660.

He, B., and Yuan, X. 2012. On the $o(1/n)$ convergence rate of the Douglas-Rachford alternating direction method. *SIAM Journal on Numerical Analysis* 50(2):700–709.

Hong, M.; Luo, Z.-Q.; and Razaviyayn, M. 2016. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization* 26(1):337–364.

Lee, D. D., and Seung, H. S. 2001. Algorithms for non-negative matrix factorization. In *NIPS*, 556–562.

Lichman, M. 2013. UCI machine learning repository.

Lin, Z.; Chen, M.; and Ma, Y. 2010. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*.

Liu, R.; Lin, Z.; and Su, Z. 2013. Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning. In *ACML*, 116–132.

Lu, C.-Y.; Min, H.; Zhao, Z.-Q.; Zhu, L.; Huang, D.-S.; and Yan, S. 2012. Robust and efficient subspace segmentation via least squares regression. *ECCV* 347–360.

Lu, C.; Zhu, C.; Xu, C.; Yan, S.; and Lin, Z. 2015. Generalized singular value thresholding. In *AAAI*, 1805–1811.

Lu, C.; Tang, J.; Yan, S.; and Lin, Z. 2016. Nonconvex nonsmooth low rank minimization via iteratively reweighted nuclear norm. *TIP* 25(2):829–839.

Lu, C.; Feng, J.; Yan, S.; and Lin, Z. 2017. A unified alternating direction method of multipliers by majorization minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Lu, C.; Yan, S.; and Lin, Z. 2016. Convex sparse spectral clustering: Single-view to multi-view. *TIP* 25(6):2833–2843.

Mairal, J. 2013. Optimization with first-order surrogate functions. In *ICML*, 783–791.

Martinez, A. M. 1998. The AR face database. *CVC Technical Report* 24.

Nesterov, Y. 2005. Smooth minimization of non-smooth functions. *Mathematical programming* 103(1):127–152.

Ng, A. Y.; Jordan, M. I.; Weiss, Y.; et al. 2002. On spectral clustering: Analysis and an algorithm. *NIPS* 2:849–856.

Rockafellar, R. T., and Wets, R. J.-B. 2009. *Variational analysis*, volume 317. Springer Science & Business Media.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *TPAMI* 22(8):888–905.

Sim, T.; Baker, S.; and Bsat, M. 2002. The CMU pose, illumination, and expression (PIE) database. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 46–51. IEEE.

Von Luxburg, U. 2007. A tutorial on spectral clustering. *Statistics and computing* 17(4):395–416.

Wang, Y.; Yin, W.; and Zeng, J. 2015. Global convergence of ADMM in nonconvex nonsmooth optimization. *arXiv preprint arXiv:1511.06324*.