Results in Control and Optimization 2 (2021) 100009

Contents lists available at ScienceDirect

Results in Control and Optimization

journal homepage: www.elsevier.com/locate/rico

Nonconvex generalization of Alternating Direction Method of Multipliers for nonlinear equality constrained problems

Junxiang Wang *, Liang Zhao

Emory University, 201 Dowman Dr, Atlanta, GA, 30322, USA

ARTICLE INFO

Keywords:
Nonconvex ADMM
Nonlinear equality constraints
Spherical constraints
Multi-instance learning

ABSTRACT

The classic Alternating Direction Method of Multipliers (ADMM) is a popular framework to solve linear-equality constrained problems. In this paper, we extend the ADMM naturally to nonlinear equality-constrained problems, called neADMM. The difficulty of neADMM is to solve nonconvex subproblems. We provide globally optimal solutions to them in two important applications. Experiments on synthetic and real-world datasets demonstrate excellent performance and scalability of our proposed neADMM over existing state-of-the-start methods.

1. Introduction

There is a growing demand for efficient computational methods for analyzing high-dimensional large-scale data across a wide variety of applications, including healthcare, finance, social media, astronomy, and e-commerce [1–3]. The classic Alternating Direction Method of Multipliers (ADMM) has received a significant amount of attention in the last few years. Its main advantage consists in the ability to split a complex problem into a series of simpler *subproblems*, each of which is easy to solve [4]. While ADMM focuses on optimization problems with linear equality constraints, many real-world problems require nonlinear constraints such as collaborative filtering [5], 1-bit compressive sensing [6], and mesh processing [7] and as yet, there lacks a discussion on how to apply ADMM to *nonlinear equality-constrained* problems.

In this paper, we extend the ADMM into the nonlinear equality-constrained problems in Section 2, called neADMM. Two important applications of neADMM are discussed in Section 3, along with a consideration of ways to solve nonconvex subproblems. Section 4 presents experiments conducted to show the effectiveness of the proposed neADMM on both synthetic and real-world datasets. We summarize this paper by Section 5.

2. Nonlinear equality-constrained ADMM

We consider the following nonconvex problem with vector variables $x_1 \in \mathbb{R}^{m_1}$ and $x_2 \in \mathbb{R}^{m_2}$.

Problem 1.

$$\min_{x_1,x_2} \; F_1(x_1) + F_2(x_2) \; s.t. \; f_1(x_1) + f_2(x_2) = 0$$

In Problem 1, $F_1(x_1)$ and $F_2(x_2)$ are proper continuous functions, and $f_1 : \mathbb{R}^{m_1} \to \mathbb{R}^d$ and $f_2 : \mathbb{R}^{m_2} \to \mathbb{R}^d$ can be nonlinear. We present the neADMM algorithm to solve Problem 1. According to the standard ADMM routine, we formulate the augmented

* Corresponding author.
E-mail addresses: jwan936@emory.edu (J. Wang), lzhao413@emory.edu (L. Zhao).

https://doi.org/10.1016/j.rico.2021.100009
Received 18 December 2020; Received in revised form 12 March 2021; Accepted 16 March 2021
Available online 24 March 2021

Lagrangian as follows:

$$L_\rho(x_1, x_2, y) = F_1(x_1) + F_2(x_2) + y^T(f_1(x_1) + f_2(x_2)) + (\rho/2)\|f_1(x_1) + f_2(x_2)\|_2^2 \tag{1}$$

where $\rho > 0$ is a penalty parameter and y is a dual variable. neADMM aims to optimize the following two subproblems alternately:

$$x_1^{k+1} = \arg\min_{x_1} L_\rho(x_1, x_2^k, y^k) \tag{2}$$

$$x_2^{k+1} = \arg\min_{x_2} L_\rho(x_1^{k+1}, x_2, y^k) \tag{3}$$

Without loss of generality, we implicitly assume that there exist minima in Eqs. (2) and (3).

The neADMM algorithm is presented in Algorithm 1. Specifically, Lines 3–4 update Eqs. (2) and (3), Line 5 updates the dual variable y, and Lines 6 and 7 update the primal residual r and the dual residual s, respectively.

The main challenge of the neADMM framework is to solve nonconvex subproblems Eqs. (2) and (3). While there is no general method to solve them exactly, for some specific forms, we have efficient solutions, which are discussed in the next section.

Algorithm 1 the neADMM Algorithm

1: Initialize x_1 and x_2, y, ρ, $k = 0$.
2: **repeat**
3: Update x_1^{k+1} in Eq. (2).
4: Update x_2^{k+1} in Eq. (3).
5: Update $y^{k+1} \leftarrow y^k + \rho(f_1(x_1^{k+1}) + f_2(x_2^{k+1}))$
6: Update $r^{k+1} \leftarrow f_1(x_1^{k+1}) + f_2(x_2^{k+1})$. # Calculate the primal residual.
7: Update $s^{k+1} \leftarrow \rho\partial f_1(x_1^{k+1})^T(f_2(x_2^{k+1}) - f_2(x_2^k))$. # Calculate the dual residual.
8: $k \leftarrow k + 1$.
9: **until** convergence.
10: Output x_1 and x_2.

3. Applications

3.1. Optimization problems with spherical constraints

The spherical constraint is widely applied in 1-bit compressive sensing [6] and mesh processing [7], which is formulated as follows:

Problem 2 (*Spherical Constrained Problem*)**.**

$$\min_x \ \ell(x), \quad s.t.\ \|x\|_2^2 = 1$$

where $\ell(\bullet)$ is a loss function. We introduce an auxiliary variable w and reformulate this problem as follows:

$$\min_{x,w} \ell(x), \quad s.t.\ \|w\|_2^2 = 1, w = x$$

The augmented Lagrangian is formulated as follows according to Eq. (1):

$$\begin{aligned} L_\rho(x, w, y_1, y_2) &= \ell(x) + y_1^T(\|w\|_2^2 - 1) + (\rho/2)\|\|w\|_2^2 - 1\|_2^2 + y_2^T(w - x) + (\rho/2)\|w - x\|_2^2 \\ &= \ell(x) + (\rho/2)\|\|w\|_2^2 - 1 + y_1/\rho\|_2^2 + (\rho/2)\|w - x + y_2/\rho\|_2^2 - \|y_1\|_2^2/(2\rho) - \|y_2\|_2^2/(2\rho). \end{aligned}$$

Due to space limit, the algorithm to solve Problem 2 is shown in Algorithm 2 in Appendix. All subproblems are detailed as follows:

1. Update x.

The variable x is updated as follows:

$$x^{k+1} \leftarrow \arg\min_x \ell(x) + (\rho/2)\|w^k - x + y_2^k/\rho\|_2^2 \tag{4}$$

This subproblem is convex and can be solved using the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [8].

2. Update w.

The variable w is updated as follows:

$$w^{k+1} \leftarrow \arg\min_w \|w - x^{k+1} + y_2^k/\rho\|_2^2 + \|\|w\|_2^2 - 1 + y_1^k/\rho\|_2^2 \tag{5}$$

This subproblem is non-convex, but we have figured out a closed-form solution, as shown in the following theorem.

Theorem 1. *The solution to Eq. (5) is*

$$w = (x^{k+1} - y_2^k/\rho)/(2\|w\|_2^2 - 1 + 2y_1^k/\rho) \tag{6}$$

where $u = \|w\|_2$ is obtained uniquely from one real root of the following cubic equation.

$$\left|2u^3 - u + 2uy_1^k/\rho\right| = \|x^{k+1} - y_2^k/\rho\|_2 \tag{7}$$

Due to space limit, its proof is shown in Appendix A.1 in Appendix.

3.2. Optimization problems with logical constraints with "max" operations

In the machine learning community, many minimization problems contain the "max" operator. For example, in the multi-instance learning problem [9], each "bag" can contain multiple "instances" and the classification task is to predict the labels of both bags and their instances. Here, a conventional logic between the ith bag and its instances is that "the bag label q_i is 1 when at least one of its instances" labels is 1; otherwise, the bag label q_i is 0". This well-known rule is referred to as the "max rule" [2], namely $q_i = \max_{j=1,\dots,n_i} t_{i,j}$, where $t_{i,j}$ is the label of the jth instance of q_i, n_i is the number of instances, and q_i and $t_{i,j}$ can be generalized to real numbers [9]. This problem can be formulated mathematically as follows:

Problem 3 (*Multi-instance Learning Problem*).

$$\min_{t,q,\beta} \ell(q) + \Omega(\beta)$$
$$s.t.\ q_i = \max_{j=1,\dots,n_i} t_{i,j},\ t_{i,j} = X_{i,j}\beta$$

where $\ell(\bullet)$ and $\Omega(\bullet)$ are the loss function and the regularization term, respectively. β is a feature weight vector, $X_{i,j}$ and $t_{i,j}$ are the jth input instance and the predicted value in the ith bag, and n_i is the number of instances in the ith bag. Let $q = [q_1, \dots, q_n]$, $t = [t_1, \dots, t_n]$ and $X = [X_1, \dots, X_n]$, where n is the number of bags, $t_i = [t_{i,1}, \dots, t_{i,n_i}]$, and $X_i = [X_{i,1}, \dots, X_{i,n_i}]$.

The augmented Lagrangian is formulated as follows according to Eq. (1):

$$\begin{aligned} L_\rho(q,\beta,t,y_1,y_2) &= \ell(q) + \Omega(\beta) + y_1^T(q - \max t) + (\rho/2)\|q - \max t\|_2^2 + y_2^T(t - X\beta) + (\rho/2)\|t - X\beta\|_2^2 \\ &= \ell(q) + \Omega(\beta) + (\rho/2)\|q - \max t + y_1/\rho\|_2^2 + (\rho/2)\|t - X\beta + y_2/\rho\|_2^2 - \|y_1\|_2^2/(2\rho) - \|y_2\|_2^2/(2\rho) \end{aligned}$$

Due to space limit, the algorithm to solve Problem 3 is shown in Algorithm 3 in Appendix. All subproblems are shown as follows:

1. Update q.

The variable q is updated as follows:

$$q^{k+1} \leftarrow \arg\min_q \ell(q) + (\rho/2)\|q - \max t^k + y_1^k/\rho\|_2^2 \tag{8}$$

This subproblem is convex and solved using FISTA [8].

2. Update β.

The variable β is updated as follows:

$$\beta^{k+1} \leftarrow \arg\min_\beta \Omega(\beta) + (\rho/2)\|t^k - X\beta + y_2^k/\rho\|_2^2 \tag{9}$$

This subproblem is convex and solved using FISTA [8].

3. Update t.

The variable t is updated as follows:

$$t^{k+1} \leftarrow \arg\min_t \|q^{k+1} - \max t + y_1^k/\rho\|_2^2 + \|t - X\beta^{k+1} + y_2^k/\rho\|_2^2 \tag{10}$$

This subproblem is nonconvex and difficult to solve. Here we apply a linear search to solve it. Due to the separability of t, we have

$$\begin{aligned} &t_i = \arg\min_{t_i} h(t_i) \\ &h(t_i) = \|q_i^{k+1} - t_{i,j^*} + y_{1,i}^k/\rho\|_2^2 + \|t_i - \varphi_i\|_2^2 \\ &s.t.\ t_{i,j^*} = \max_{j=1,\dots,n_i} t_{i,j} \end{aligned} \tag{11}$$

where $\varphi_{i,j} = X_{i,j}^T\beta^{k+1} - y_{2,i,j}^k/\rho$ is constant.

It is easy to find that $t_{i,j} = \min(t_{i,j^*}, \varphi_{i,j}) \le t_{i,j^*}$. We need to consider two cases: (1) $t_{i,j} < t_{i,j^*}$, (2) $t_{i,j} = t_{i,j^*}$. This problem is therefore split into two subroutines: (1) find solutions to two cases, and (2) decide which case every instance belongs to.

Subroutine 1. For case (1), we have a closed-form solution $t_{i,j} = \varphi_{i,j}$. For case (2), we define a set $C = \{j : t_{i,j} = t_{i,j^*}\}$ and its complement $\overline{C} = n_i - C$, and we plug it in Eq. (11) to obtain:

$$t_{i,j^*} = \Big(\sum_{j\in C} \varphi_{i,j} + q_i^{k+1} + y_{1,i}^k/\rho\Big)/(|C| + 1).$$

Subroutine 2. Solving case (2) is equivalent to minimizing $h(t_i)$ by selecting appropriate indexes for set C. The definition of C implies that C consists of indexes that have the largest $\varphi_{i,j}$. Otherwise, $t_{i,j} < t_{i,j^*}$ and hence $i \notin C$. Now we need to determine how many indexes C should have. Let φ_i' be a decreasing order of φ_i, $|C| = c$ and $a_{i,c} = (\sum_{j=1}^c \varphi_{i,j}' + q_i^{k+1} + y_{1,i}^k/\rho)/(c+1)$. then we have the following theorem:

Theorem 2. *$h(t_i)|_{t_{i,j^*}=a_{i,c}}$ increases monotonically with c, where $h(t_i)$ is defined in Eq. (11).*

Due to space limit, its proof is in Appendix A.2 in Appendix. The above theorem implies that the smallest c minimizes Eq. (11). So the objective becomes

$$c^* = \arg\min_c c,\ s.t.\ a_{i,c} > \varphi_{i,c+1}'$$

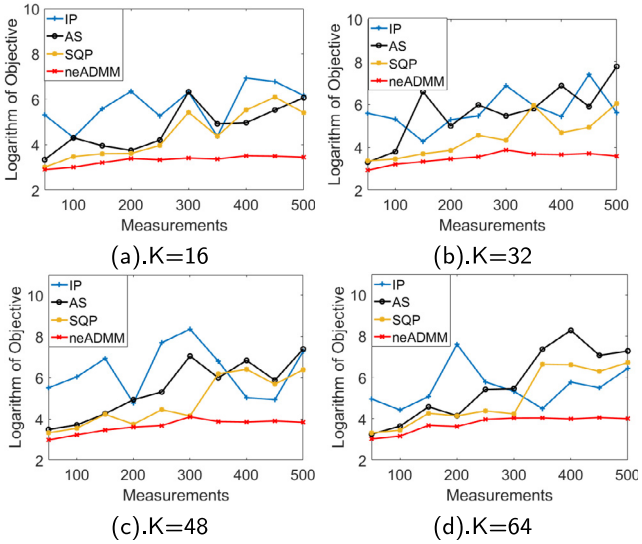

Fig. 1. Measurements versus objective values for different choices of K: the objective of the neADMM is the smallest.

The corresponding solution is:

$$t_{i,j} = min(\varphi_{i,j}, a_{i,c^*})$$

The time complexity is $O(n_i log n_i)$ because the main operation of the linear search is to sort φ_i in decreasing order.

4. Experiments

In this section, we assess the performance of our proposed neADMM on two applications.[1] The experiments were conducted on a 64-bit machine equipped with an InteLoss(R) core(TM) processor (i7-6820HQ CPU) and 16.0GB memory.

4.1. 1-bit compressive sensing

4.1.1. Problem settings

In Problem 2, we set $\ell(x) = \|x\|_1 + \lambda/2 \sum \min(Y\phi x, 0)^2$ where $\phi \in \mathbb{R}^{M\times N}$ is a measurement operator, $Y \in \mathbb{R}^{M\times M}$ is a measurement matrix and $\lambda > 0$ is a tuning parameter. Here, N represents the number of signals, M denotes the number of measurements and K denotes the number of nonzero signals. λ and ρ were set to 0.01 and 1, respectively, and the maximal number of iterations was set to 100. The comparison methods were the Interior Point (IP) method [10], the Active Set (AS) method [11], and the Sequential Quadratic Programming (SQP) method [12]. They were all provided by the Matlab optimization toolbox and shared the same initial points.

4.1.2. Performance

Fig. 1 shows the relationship between the number of measurements and objective values for different choices of K. Overall, the objective values of the neADMM are lower (i.e. better) than those of the comparison methods. The logarithms of the objective values of the neADMM are all around 3, while these of comparison methods fluctuate somewhat. Even though all comparison methods are state-of-the-art, our proposed neADMM performs better maybe due to its inherent splitting schemes: all subproblems of neADMM have global optima, which make the neADMM easier to find better solutions, while all comparison methods may plunge into saddle points or local minima.

4.2. Multi-instance learning

This experiment validates the effectiveness of our proposed neADMM against several comparison methods for multi-instance learning problems.

[1] Our code is available at https://github.com/xianggebenben/neADMM.

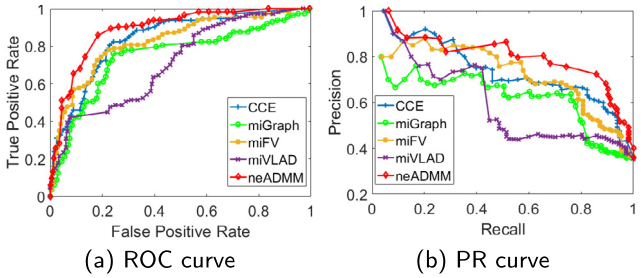

Fig. 2. The ROC and PR curves for all methods on the vaccine adverse events dataset: the neADMM outperforms others.

Table 1
Model performance on the two datasets under six metrics.

Vaccine Adverse Events						
Method	ACC	PR	RE	FS	AUC	AUPR
CCE	0.7397	0.7818	0.3805	0.5119	0.8401	0.7325
miGraph	0.7206	0.6812	0.4159	0.5165	0.7465	0.6128
miFV	0.6603	0.80	0.0708	0.1301	0.8266	0.7136
miVLAD	0.7365	0.7419	0.4071	0.5257	0.7110	0.5978
neADMM	**0.8000**	**0.8049**	**0.5841**	**0.6769**	**0.8901**	**0.7961**
Fox Images						
Method	ACC	PR	RE	FS	AUC	AUPR
CCE	**0.5250**	0.4444	0.4706	0.4571	0.5358	0.4250
miGraph	0.4250	0.4250	**1.0000**	**0.5965**	0.4425	0.3824
miFV	0.4500	0.4000	0.5882	0.4762	0.4757	0.3760
miVLAD	0.4500	0.4242	0.8235	0.5600	0.5192	0.4288
neADMM	0.5000	**0.4483**	0.7647	0.5652	**0.5422**	**0.4583**

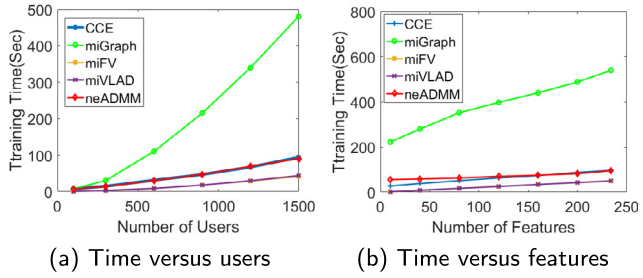

Fig. 3. The running time for all the methods for the vaccine adverse events dataset: the running time increases approximately linearly with the number of users and features.

4.2.1. Problem settings

Two datasets are used for the performance evaluation: vaccine adverse events [3] and fox images [13]. For each dataset, 80% are trained with a classifier and the remaining 20% used for testing. Four comparison methods are Constructive Clustering based Ensemble (CCE) [14], Multi-instance learning with graph (miGraph) [15], Multi-instance Learning based on the Vector of Locally Aggregated Descriptors representation (miVLAD) [16], and Multi-instance Learning based on the Fisher Vector representation (miFV) [16]. We set the loss function $L(\bullet)$ to be a logarithm loss. $\Omega(\beta) = \lambda\|\beta\|_1$ where $\lambda > 0$ is a regularization parameter and was set to 1. The penalty parameter ρ was set to 0.1, and the maximal number of iterations was set to 100. Six metrics were utilized: Accuracy (ACC), Precision (PR), Recall (RE), F-score (FS), Area Under Receiver Operating Characteristic curve (AUC) , and Area Under Precision–Recall curve (AUPR). More details can be found in Appendix.

4.2.2. Performance

Table 1 summarizes the prediction results obtained using neADMM and the four comparison methods on two datasets. In general, the metrics for neADMM are better than those for any of the comparison methods including for AUC and AUPR, which are the most important metrics.

Fig. 2 shows that the ROC and PR curves of all the comparison methods are surrounded by these of neADMM, which is consistent with the data shown in Table 1.

4.2.3. Scalability analysis

To examine the scalability of neADMM, we measure the running time for all methods, by averaging twenty times. As shown in Fig. 3, the running time of all the methods increases linearly with the number of users and features, and The proposed neADMM is computationally efficient.

5. Conclusions

We propose neADMM, an extension of ADMM framework for nonlinear equality-constrained problems. The challenge of neADMM is to solve nonconvex subproblems. Our main contribution is to provide solutions to them in two specific applications: in the spherical-constrained problem, one nonconvex subproblem is solved by roots of a cubic equation (Theorem 1); for the problem with "max" operations, we solve its nonconvex subproblem by a linear search (Theorem 2). We theoretically guarantee that solutions to these two subproblems are globally optimal. Experiments demonstrate that our proposed neADMM performs the best, and scales well with the increase of features and samples.

In the future, we may explore the convergence guarantee of the proposed neADMM because convergence conditions in the existing literatures cannot be applied to the neADMM.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the National Science Foundation (NSF) Grant No. 1755850, No. 1841520, No. 2007716, No. 2007976, No. 1942594, No. 1907805, a Jeffress Memorial Trust Award, Amazon Research Award, NVIDIA GPU Grant, and Design Knowledge Company (subcontract number: 10827.002.120.04).

Appendix A. Theorem proofs

A.1. The proof of Theorem 1

Proof.

It is obvious that as $w \to \infty$ or $w \to -\infty$, the value of Eq. (5) approaches infinity. This indicates that there exists a global minimum in Eq. (5). In general, the globally minimal point of a function is either a point 1) whose gradient is 0 or 2) whose gradient does not exist, or 3) a point at the boundary of the domain [17]. In our case, the domain of Eq. (5) is the set of real vectors, whose boundary is empty (i.e. condition 3 is impossible), and the derivative of Eq. (5) exists for any real vector w (i.e. condition 2 is impossible). Therefore, the gradient of the globally minimal point of Eq. (5) must be 0, which leads to

$$2(w-x^{k+1}+y_2^k/\rho)+4(\|w\|_2^2-1+2y_1^k/\rho)w = 0 \tag{12}$$

Eq. (6) is obtained directly from Eq. (12). Now the only issue is to obtain $\|w\|_2$, which appears on the right side of Eq. (6). To achieve this, we obtain Eq. (7) by taking the norm on both sides of Eq. (6) as follows:

$$\|w\|_2 = \|x^{k+1}-y_2^k/\rho\|_2 / \left|2\|w\|_2^2-1+2y_1^k/\rho\right| \tag{13}$$

where $2\|w\|_2^2-1+2y_1^k/\rho$ is a scalar. Let $u = \|w\|_2$, Eq. (13) is equivalent to Eq. (7). Eq. (7) is a cubic equation with regard to u, and can be solved using Cardano's Formula. In order to obtain the unique u, we consider two possibilities of three roots of Eq. (7):

1. One pair of conjugate imaginary roots and one real root. In this case, u is the real root.
2. Three real roots. In this case, we have three possible values of w using Eq. (6), which correspond to three real roots. u is the root whose corresponding w minimizes Eq. (5).

After u is obtained, w can be obtained using Eq. (6). □

A.2. The proof of Theorem 2

Proof.

$$\begin{aligned}
&h(t_i)|_{t_{i,j^*}=a_{i,c+1}} - h(t_i)|_{t_{i,j^*}=a_{i,c}} \\
&= \sum_{j=1}^{c+1}(a_{i,c+1}-\varphi'_{i,j})^2+(a_{i,c+1}-q_i^{k+1}-y_{1,i}^k/\rho)^2 - \sum_{j=1}^{c}(a_{i,c}-\varphi'_{i,j})^2-(a_{i,c}-q_i^{k+1}-y_{1,i}^k/\rho)^2 \\
&= (a_{i,c+1}-a_{i,c})((c+1)(a_{i,c+1}+a_{i,c}) - 2\sum_{j=1}^{c}\varphi'_{i,j} - 2q_i^{k+1} - 2y_{1,i}^k/\rho) \\
&= (a_{i,c+1}-a_{i,c})(\varphi'_{i,c+1}-a_{i,c+1})
\end{aligned}$$

$$= (a_{i,c+1} - a_{i,i})((c+2)a_{i,c+1} - (c+1)a_{i,c} - a_{i,c+1})$$
$$= (c+1)(a_{i,c+1} - a_{i,c})^2 \geqslant 0.$$

Hence the theorem is proven. □

Appendix B. Algorithms to solve Problems 2 and 3

The neADMM algorithm used to solve Problem 2 is outlined in Algorithm 2. Specifically, Lines 9–10 update the dual variables y_1 and y_2, respectively, Lines 11–12 compute the primal and dual residuals, respectively, Lines 3–4 update x and w alternately. Unfortunately, Algorithm 2 is not necessarily convergent by our numeric experiments, but it outperforms existing state-of-the-art methods. The computational cost of Algorithm 2 mainly consists in Eq. (4), whose time complexity is $O(1/k^2)$, where k is the number of iterations in FISTA [8].

Algorithm 2 The Solution to Problem 2 Using neADMM

1: Initialize $x,\ w,\ y_1,\ y_2\ \rho > 0,\ k = 0$.
2: **repeat**
3: Update x^{k+1} in Eq. (4).
4: Update w^{k+1} in Eq. (5).
5: Update $r_1^{k+1} \leftarrow \|w^{k+1}\|_2^2 - 1$.
6: Update $r_2^{k+1} \leftarrow w^{k+1} - x^{k+1}$.
7: Update $s_1^{k+1} \leftarrow \rho(\|w^{k+1}\|_2^2 - \|w^k\|_2^2)$.
8: Update $s_2^{k+1} \leftarrow \rho(w^{k+1} - w^k)$.
9: Update $y_1^{k+1} \leftarrow y_1^k + \rho r_1^{k+1}$.
10: Update $y_2^{k+1} \leftarrow y_2^k + \rho r_2^{k+1}$.
11: Update $r^{k+1} \leftarrow \sqrt{\|r_1^{k+1}\|_2^2 + \|r_2^{k+1}\|_2^2}$. # Calculate the primal residual.
12: Update $s^{k+1} \leftarrow \sqrt{\|s_1^{k+1}\|_2^2 + \|s_2^{k+1}\|_2^2}$. # Calculate the dual residual.
13: $k \leftarrow k + 1$.
14: **until** convergence.
15: Output x and w.

Algorithm 3 The Solution to Problem 3 Using neADMM

1: Initialize $q,\ \beta,\ t,\ y_1,\ y_2,\ \rho > 0,\ k = 0$.
2: **repeat**
3: Update q^{k+1} in Eq. (8).
4: Update β^{k+1} in Eq. (9).
5: Update t^{k+1} in Eq. (10).
6: Update $r_1^{k+1} \leftarrow q^{k+1} - \max t^{k+1}$.
7: Update $r_2^{k+1} \leftarrow t^{k+1} - X\beta^{k+1}$.
8: Update $s_1^{k+1} \leftarrow \rho(\max t^k - \max t^{k+1})$.
9: Update $s_2^{k+1} \leftarrow t^{k+1} - t^k$.
10: Update $y_1^{k+1} \leftarrow y_1^k + \rho r_1^{k+1}$.
11: Update $y_2^{k+1} \leftarrow y_2^k + \rho r_2^{k+1}$.
12: Update $r^{k+1} \leftarrow \sqrt{\|r_1^{k+1}\|_2^2 + \|r_2^{k+1}\|_2^2}$. # Calculate the primal residual.
13: Update $s^{k+1} \leftarrow \sqrt{\|s_1^{k+1}\|_2^2 + \|s_2^{k+1}\|_2^2}$. # Calculate the dual residual.
14: $k \leftarrow k + 1$.
15: **until** convergence.
16: Output $q,\ \beta$ and t.

The neADMM Algorithm to solve Problem 3 is stated in Algorithm 3. Specifically, Lines 12–13 compute the residuals, Lines 10 and 11 update the dual variables y_1 and y_2, respectively, and Lines 3–5 update q, β and t alternately.

Appendix C. More experimental details of multi-instance learning

C.1. Comparison methods

The following methods serve as baselines for the performance comparison.

1. Constructive Clustering-based Ensemble (CCE) [14]. Each instance in the bag is first clustered into groups, then a classifier distinguishes a bag from the others based on group information. Many classifiers are generated due to the many different group numbers. The final step is then to gather all the classifiers together.

2. Multi-instance learning with graph (miGraph) [15]. The miGraph treats the instances in the bags as non-independently and identically distributed. It then implicitly constructs a graph by considering the affinity matrices and defines a new graph kernel which contains the clique information.

3. Multi-instance Learning based on the Vector of Locally Aggregated Descriptors representation (miVLAD) [16]. Here, multiple instances are mapped into a high dimensional vector by the Vector of Locally Aggregated Descriptors (VLAD) representation. The SVM can then be applied to train a classifier.

4. Multi-instance Learning based on the Fisher Vector representation (miFV) [16]. The miFV was similar to the miVLAD except that multiple instances were encoded by the Fisher Vector (FV) representation.

C.2. Metrics

This experiment utilizes six metrics to evaluate model performance: the Accuracy (ACC) is the ratio of accurately labeled bags to all bags; the Precision (PR) is the ratio of those accurately labeled as positive bags to all those labeled as positive bags; the Recall (RE) defines the ratio of those accurately labeled as positive bags to all positive bags; the F-score (FS) is the harmonic mean of the precision and recall; the Receiver Operating Characteristic curve (ROC curve) and the Precision–Recall curve (PR curve) both delineate the classification ability of a model as its discrimination threshold varies; the Area Under ROC (AUC) and the Area Under PR curve (AUPR) are the most important metrics when evaluating the performance of a classifier.

Appendix D. Related work

In this section, we summarize the existing works in this area.

Nonconvex ADMM. Most research on linear ADMM has focused on the conditions needed to guarantee convergence. For example, Boyd et al. proved the convergence of the ADMM with two variables under mild conditions [4]. The next step was multi-block ADMM, which refers to ADMM with at least three variables, which was first studied by Chen et al. [18] who concluded that the multi-block ADMM does not necessarily converge by providing a counterexample; He and Yuan explained why this is the case from the perspective of a variational inequality framework [19]. Since then, many researchers have sought to define the sufficient conditions required to ensure the convergence of multi-block ADMM. For example, Robinson and Tappenden [20] and Lin et al. [21] imposed a strong convexity assumption on the objective function. There has been considerable interest in extending the multi-block ADMM into nonconvex settings; see [22–28] for more information.

Aside from convergence contributions, a handful of works have focused on applying ADMM to address real-world problems. For example, Zhang discussed the nonnegative matrix factorization problem [29]; Wahlberg et al. presented an ADMM algorithm to deal with the total variance estimation problem [30], and Yang and Zhang explored the possibility of utilizing ADMM for sparse solution recovery in compressive sensing [31]; Lin et al. proved that the multi-block ADMM converged in the regularized least-squares decomposition problem [32]; while Wang and Zhao considered the application of multi-block ADMM to solve isotonic regression problems [33].

Optimization problems with nonlinear constraints. A few papers have studied problems with nonlinear constraints. For instance, Julier and LaViola applied a Kalman filter-type estimator to describe the state space of some physical systems using a projection method [34]; Lawrence and Tits proposed a simple scheme to deal with nonlinear constraints in the context of sequential quadratic programming [35]; and Celis et al. applied a trust-region method to solve nonlinear-equality constrained problems [36].

References

[1] Baniasadi P, Foumani M, Smith-Miles K, Ejov V. A transformation technique for the clustered generalized traveling salesman problem with applications to logistics. European J Oper Res 2020;285(2):444–57.
[2] Wang J, Zhao L. Multi-instance domain adaptation for vaccine adverse event detection. In: Proceedings of the 2018 world wide web conference on world wide web. 2018. p. 97–06.
[3] Wang J, Zhao L, Ye Y. Semi-supervised multi-instance interpretable models for flu shot adverse event detection. In: 2018 IEEE international conference on big data. IEEE; 2018, p. 851–60.
[4] Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning 2011;3(1):1–122.
[5] Candes E, Recht B. Exact matrix completion via convex optimization. Commun ACM 2012;55(6):111–9.
[6] Boufounos PT, Baraniuk RG. 1-bit compressive sensing. In: Information sciences and systems, 2008. 42nd annual conference on. IEEE; 2008, p. 16–21.
[7] Neumann T, Varanasi K, Theobalt C, Magnor M, Wacker M. Compressed manifold modes for mesh processing. In: Computer graphics forum, vol. 33. Wiley Online Library; 2014, p. 35–44.
[8] Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J Imag Sci 2009;2(1):183–202.
[9] Amores J. Multiple instance classification: Review, taxonomy and comparative study. Artificial Intelligence 2013;201:81–105.
[10] Byrd RH, Hribar ME, Nocedal J. An interior point algorithm for large-scale nonlinear programming. SIAM J Optim 1999;9(4):877–900.
[11] Nocedal J, Wright S. Numerical optimization. Springer Science & Business Media; 2006.
[12] Fletcher R. Practical methods of optimization. John Wiley & Sons; 2013.
[13] Andrews S, Tsochantaridis I, Hofmann T. Support vector machines for multiple-instance learning. In: Advances in neural information processing systems. 2003, p. 577–84.
[14] Zhou Z-H, Zhang M-L. Solving multi-instance problems with classifier ensemble based on constructive clustering. Knowl Inf Syst 2007;11(2):155–70.
[15] Zhou Z-H, Sun Y-Y, Li Y-F. Multi-instance learning by treating instances as non-iid samples. In: Proceedings of the 26th annual international conference on machine learning. ACM; 2009, p. 1249–56.
[16] Wei XS, Wu J, Zhou ZH. Scalable multi-instance learning. In: IEEE international conference on data mining. 2014. p. 1037–42.
[17] Fitzpatrick P. Advanced calculus, vol. 5. American Mathematical Soc.; 2009.
[18] Chen C, He B, Ye Y, Yuan X. The direct extension of admm for multi-block convex minimization problems is not necessarily convergent. Mathe Programm 2016;155(1-2):57–79.
[19] He B, Yuan X. On the o(1/n) convergence rate of the douglas–rachford alternating direction method. SIAM J Numer Anal 2012;50(2):700–9.
[20] Robinson DP, Tappenden R. A flexible admm algorithm for big data applications. J Sci Comput 2017;71(1):435–67.
[21] Lin T-Y, Ma S-Q, Zhang S-Z. On the sublinear convergence rate of multi-block admm. J Oper Res Soc China 2015;3(3):251–74. http://dx.doi.org/10.1007/s40305-015-0092-0, https://doi.org/10.1007/s40305-015-0092-0.

[22] Latorre F, Cevher V, et al. Fast and provable admm for learning with generative priors. In: Advances in Neural Information Processing Systems. 2019, p. 12004–16.
[23] Magnússon S, Weeraddana PC, Rabbat MG, Fischione C. On the convergence of alternating direction lagrangian methods for nonconvex structured optimization problems. IEEE Trans Control Netw Syst 2016;3(3):296–309.
[24] Wang F, Xu Z, Xu H-K. Convergence of bregman alternating direction method with multipliers for nonconvex composite problems. Sci China Inform Sci 2018. http://dx.doi.org/10.1007/s11432-017-9367-6.
[25] Wang J, Chai Z, Cheng Y, Zhao L. Toward model parallelism for deep neural network based on gradient-free admm framework. In: Proceedings of the 20th IEEE International Conference on Data Mining, ICDM 2020. Sorrento, Italy; 2020.
[26] Wang J, Yu F, Chen X, Zhao L. Admm for efficient deep learning with global convergence. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA; 2019.
[27] Wang J, Zhao L, Wu L. Multi-convex inequality-constrained alternating direction method of multipliers. 2019.
[28] Wang Y, Yin W, Zeng J. Global convergence of admm in nonconvex nonsmooth optimization. J Sci Comput 2015;1–35.
[29] Zhang Y. An alternating direction algorithm for nonnegative matrix factorization, preprint. Citeseer. 2010.
[30] Wahlberg B, Boyd S, Annergren M, Wang Y. An admm algorithm for a class of total variation regularized estimation problems. IFAC Proceedings Volumes 2012;45(16):83–8.
[31] Yang J, Zhang Y. Alternating direction algorithms for ℓ_1-problems in compressive sensing. SIAM j Sci Comput 2011;33(1):250–78.
[32] Lin T, Ma S, Zhang S. Global convergence of unmodified 3-block admm for a class of convex minimization problems. J Sci Comput 2018;76(1):69–88.
[33] Wang J, Zhao L. The application of multi-block admm on isotonic regression problems. Workshop Optim Machine Learn 2019.
[34] Julier SJ, LaViola JJ. On kalman filtering with nonlinear equality constraints. IEEE Trans Signal Process 2007;55(6):2774–84.
[35] Lawrence CT, Tits AL. Nonlinear equality constraints in feasible sequential quadratic programming. Optim Methods Softw 1996;6(4):265–82.
[36] Celis M, Dennis J, Tapia R. A trust region strategy for nonlinear equality constrained optimization. Numer Optim 1985;1984:71–82.