Convergence and Applications of ADMM on the Multi-convex Problems

Junxiang Wang and Liang Zhao

Department of Computer Science and Informatics, Emory University

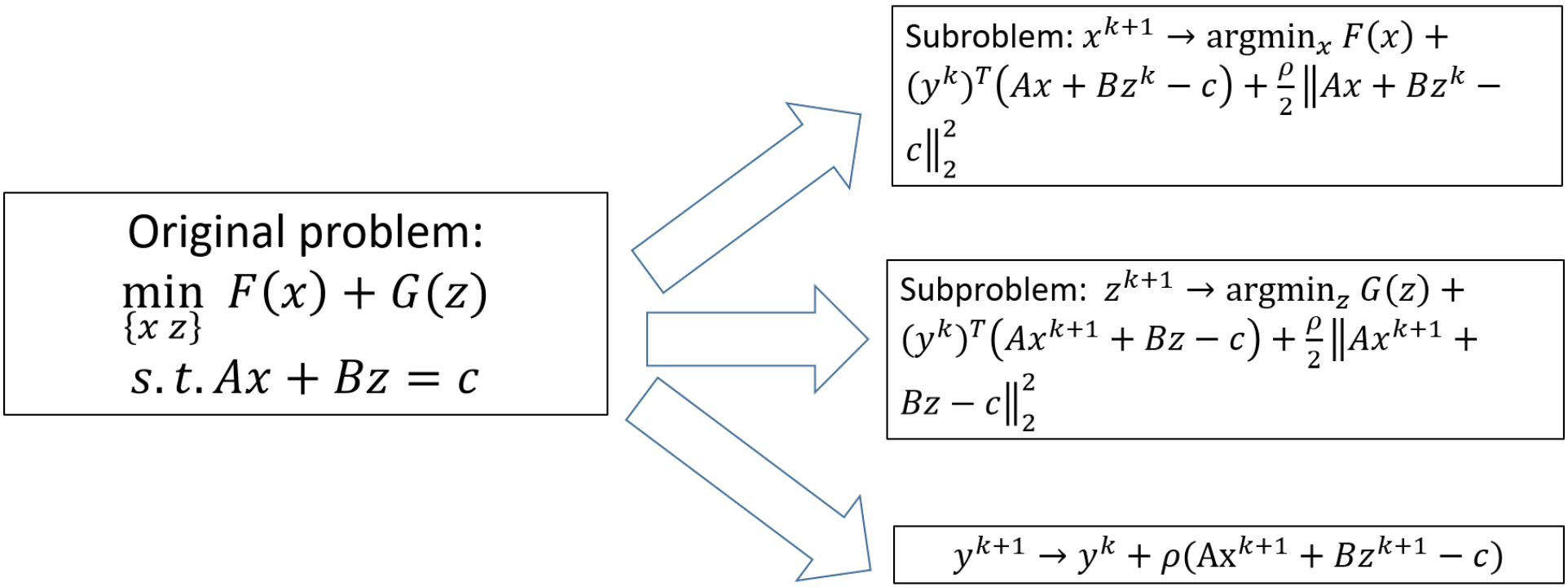
The 26th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2022)

Table of Contents

Table of Contents

ADMM Formulation

The Alternating Direction Method of Multipliers(ADMM) has received great attention in recent years. The mechanism of ADMM is to split an optimization problem into multiple subproblems, each of which is easy to solve.

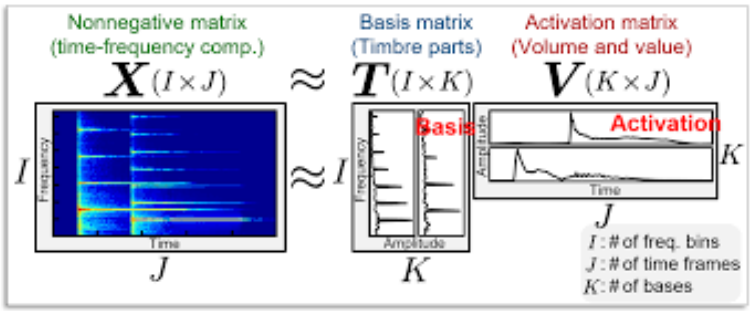


ADMM Formulation

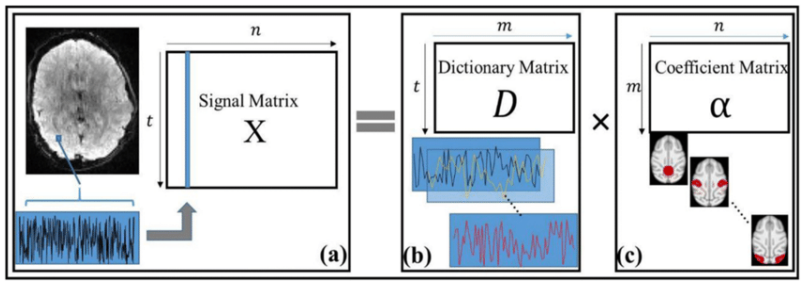
where $F(x)$ and $G(z)$ are convex, A and B are matrices and c is a vector.

Multi-convex Problems

There has been an increasing number of real-world applications where objective functions are multi-convex (i.e. nonconvex for all variables but convex for each when all others are fixed).



(a). Nonnegative Matrix Factorization



(b). Sparse Dictionary Learning

Figure: Two Applications of Multi-convex Problems

ADMM on Multi-convex Problems

However, there is no investigation on ADMM to solve multi-convex problems. Specifically, two questions are needed to answer:

- What conditions are required to ensure the convergence of ADMM on multi-convex problems?
- What multi-convex applications can be addressed by ADMM?

This motives us to investigate both theories and applications of multi-convex ADMM in this paper.

Table of Contents

Formulation of the Multi-convex Problem

The focus of this paper is the following:

$$\min_{x_1,\cdots,x_n,z} F(x_1,\cdots,x_n,z) = f(x_1,\cdots,x_n) + h(z)$$
$$s.t. \sum_{i=1}^{n} A_i x_i - z = 0.$$

where f is a proper, continuous, multi-convex and a possibly nonsmooth function, h is a proper, differentiable and convex function. $A_i (i = 1, \cdots, n)$ are matrices. This is a nonconvex problem because f is nonconvex. Without loss of generality, we assume that $F(x_1, \cdots, x_n, z)$ is lower bounded from below. The assumption of the multi-convex problem:

Assumption (Lipschitz Differentiability)

$h(z)$ is Lipschitz differentiable with constant $H \geq 0$.

Most loss functions such as the cross-entropy loss and the square loss are Lipschitz differentiable.

The Augmented Lagrangian Function

The augmented Lagrangian function is formulated mathematically as follows:

$$L_\rho(x_1, \cdots, x_n, z, y) = F(x_1, \cdots, x_n, z) + y^T(\sum_{i=1}^n A_i x_i - z) + (\rho/2)\|\sum_{i=1}^n A_i x_i - z\|_2^2.$$

where y is a dual variable and $\rho > 0$ is a penalty parameter. The proposed ADMM aims to optimize the following $n+1$ subproblems alternately.

$$x_i^{k+1} \leftarrow \arg\min_{x_i} f(\cdots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \cdots) + (y^k)^T A_i x_i + (\rho/2)\|\sum_{j=1}^{i-1} A_j x_j^{k+1} + A_i x_i + \sum_{j=i+1}^n A_j x_j^k - z^k\|_2^2. \quad (1)$$

$$z^{k+1} \leftarrow \arg\min_z L_\rho(\cdots, x_n^{k+1}, z, y^k) \quad (2)$$

$$= \arg\min_z h(z) - (y^k)^T z + (\rho/2)\|\sum_{i=1}^n A_i x_i^{k+1} - z\|_2^2.$$

The Proposed ADMM Algorithm

Algorithm The Proposed ADMM to the Multi-convex Problem

Require: $A_i (i = 1, \cdots, n), \delta > 0$.
Ensure: $x_i (i = 1, \cdots, n), z$.

Initialize ρ, $k = 0$.
repeat
  for i=1 to n **do**
    Update x_i^{k+1} in Equation (1).
  end for
  Update z^{k+1} in Equation (2).
  $r^{k+1} \leftarrow \sum_{i=1}^{n} A_i x_i^{k+1} - z^{k+1}$. # update primal residual
  $y^{k+1} \leftarrow y^k + \rho r^{k+1}$.
  $k \leftarrow k + 1$.
until $\|r^{k+1}\| \leq \delta$.
Output $x_i (i = 1, \cdots, n), z$.

Table of Contents

Convergence Properties

Two convergence properties hold based on Assumption 1. The first property states that the augmented Lagrangian L_ρ keeps decreasing.

Lemma (Objective Descent)

If $\rho > 2H$ so that $C_1 = \rho/2 - H/2 - H^2/\rho > 0$, then there exists $C_2 = \min(\rho/2, C_1)$ such that

$$\begin{aligned} & L_\rho(x_1^k, \cdots, x_n^k, z^k, y^k) - L_\rho(x_1^{k+1}, \cdots, x_n^{k+1}, z^{k+1}, y^{k+1}) \\ & \geq C_2(\|z^{k+1} - z^k\|_2^2 + \sum_{i=1}^n \|A_i(x_i^{k+1} - x_i^k)\|_2^2). \end{aligned} \tag{3}$$

The second property states that the augmented Lagrangian is bounded from below.

Lemma (Objective Bound)

If $\rho > 2H$, then $L_\rho(x_1^k, \cdots, x_n^k, z^k, y^k)$ is lower bounded.

Global Convergence

Now we can prove that the proposed ADMM converges globally in the following theorem (i.e. whatever parameters are initialized).

Theorem (Residual and Objective Convergence)

If $\rho > 2H$, then for the bounded sequence $(x_1^k, \cdots, x_n^k, z^k, y^k)$, then it has the following properties:
a). Residual convergence. This means that as $k \to \infty$, $r^k \to 0$, where r^k is defined in Algorithm 1.
b). Objective convergence. This means that as $k \to \infty$, $F(x_1^k, \cdots, x_n^k, z^k)$ converges.

However, $x_i^k (i = 1, \cdots, n)$ and z^k are not necessarily shown to be convergent.

Convergence To a Nash Point

The next theorem states that any limit point is a feasible Nash Point of the multi-convex problem.

Theorem (Convergence to a Nash Point.)

Let $\rho > 2H$, if **either** *of two assumptions hold: (a). $A_i (i = 1, \cdots, n)$ have full rank. (b). F is strongly convex with regard to x_i. Then for bounded variables $(x_1^k, \cdots, x_n^k, z^k)$, it has at least a limit point $(x_1^*, \cdots, x_n^*, z^*)$, and any limit point $(x_1^*, \cdots, x_n^*, z^*)$ is a feasible Nash point of F defined in Problem 1. That is*

$$\sum A_i x_i^* - z^* = 0. \text{ (feasibility)}$$

$$F(x_1^*, \cdots, x_n^*, z^*) \leq F(x_1^*, \cdots, x_{i-1}^*, x_i, x_{i+1}^*, \cdots, x_n^*, z^*),$$

$$\forall (x_1^*, \cdots, x_{i-1}^*, x_i, x_{i+1}^*, \cdots, x_n^*, z^*) \in dom(F), (i = 1, \cdots, n).$$

$$F(x_1^*, \cdots, x_n^*, z^*) \leq F(x_1^*, \cdots, x_n^*, z) \forall (x_1^*, \cdots, x_n^*, z) \in dom(F)$$

(Nash point).

Convergence Rate

The following theorem states that our proposed ADMM can achieve a sublinear convergence rate of $o(1/k)$ under Assumption 1, despite the nonconvex and complex nature of the multi-convex problem. Such a rate is state-of-the-art even compared to those methods for simpler convex problems.

Theorem (Convergence Rate)

If $\rho > 2H$, for a bounded sequence $(x_1^k, \cdots, x_n^k, z^k, y^k)$, define $u_k = \min_{0 \le l \le k}(\|z^{l+1} - z^l\|_2^2 + \sum_{i=1}^n \|A_i(x_i^{l+1} - x_i^l)\|_2^2)$, then the convergence rate of u_k is $o(1/k)$.

Discussion

Our proposed ADMM is more general than some influential works in terms of formulation.

- When the linear constraint $\sum_{i=1}^{n} A_i x_i = z$ is removed, the ADMM is reduced to the Block Coordinate Descent (BCD).
- When $f(x_1, \cdots, x_n) = 0$, the proposed ADMM is reduced to the convex multi-block ADMM, i.e. the ADMM with no less than three variables.

Table of Contents

Weakly-constrained Multi-task Learning(1)

- In multi-task learning problems, most papers on multi-task learning enforce the assumption of similarity among the feature weight values across tasks.
- However, this assumption is usually too strong. A probably more reasonable assumption is that multiple tasks share similar polarities for the same feature.
- For example, the feature 'number of clinic visits' will be positively related to flu outbreaks, while the feature 'popularity of vaccination' will be negatively related to them.
- This is achieved by enforcing the requirement for every pair of tasks with neighboring indices to have the same weight sign.

Weakly-constrained Multi-task Learning(2)

This optimization objective is shown as follows:

$$\min_{w_1,\cdots,w_n} \sum_{i=1}^{n} (Loss_i(w_i) + \Omega_i(w_i)) \tag{4}$$
$$s.t.\ w_{i,j} w_{i+1,j} \geq 0\ (i = 1, 2, \cdots, n-1, j = 1, 2, \cdots, m).$$

where w_i is the weight of the i-th task, and $Loss_i(w_i)$ and $\Omega_i(w_i)$ are the loss function and the regularization term of the i-th task, respectively. $w_{i,j} w_{i+1,j} \geq 0$ implies that the i-th and the $i+1$-th tasks share the same sign for their weights. Equation (4) is rewritten as follows:

$$\min_{w_1,\cdots,w_n,z} \sum_{i=1}^{n} (Loss_i(w_i) + \Omega_i(z_i)) + \lambda_1 \sum_{i=1}^{n-1} \sum_{j=1}^{m} c_1(w_{i,j} w_{i+1,j})$$
$$s.t.\ z_i = w_i\ (i = 1, 2, \cdots, n).$$

where $z = [z_1; \cdots; z_n]$ is an auxiliary variable, and $\lambda_1 > 0$ is a tuning parameter. Notice that $w_{i,j} w_{i+1,y} \geq 0$ is transformed to $c_1(x)$ such that

$$c_1(x) = \begin{cases} x^2 & x < 0 \\ 0 & x \geq 0 \end{cases}.$$

Learning with Signed-Network Constraints(1)

The application of network models for social network analysis has attracted the attention of a large number of researchers. The problem with network constraints is formulated as follows:

$$\min_{\beta_1,\cdots,\beta_n} Loss(\beta_1,\cdots,\beta_n) + \sum_{i=1}^{n} \omega_i(\beta_i)$$
$$s.t.\ \exists(\beta_i,\beta_j) \in E_s, \exists(\beta_p,\beta_q) \in E_d\ (1 \le i,j,p,q \le n).$$

where β_i is the weight of the i-th node. $Loss(\beta_1,\cdots,\beta_n)$ is a loss function and $\omega_i(\beta_i)$ is a regularization term for the i-th node.
$E_s = \{(\beta_i,\beta_j)|\beta_i\beta_j \ge 0\}$ and $E_d = \{(\beta_p,\beta_q)|\beta_p\beta_q \le 0\}$ are two edge sets to represent two opposite relationships. For example, in the problem of social event forecasting with French and English, E_s and E_d are edge sets of synonyms and antonyms between French and English, and the weight pair of the French word "bien" and the English word "good" belongs to E_s.

Learning with Signed-Network Constraints(2)

The problem can be rewritten to fit into the multi-convex problem:

$$\min_{\beta_1,\cdots,\beta_n,z} Loss(\beta_1,\cdots,\beta_n) + \sum_{i=1}^{n} \omega_i(z_i) + \lambda_2\Big(\sum_{(\beta_i,\beta_j)\in E_s} c_2(\beta_i,\beta_j) + \sum_{(\beta_p,\beta_q)\in E_d} c_3(\beta_p,\beta_q)\Big) \quad s.t.\ z_i = \beta_i\ (i = 1, 2, \cdots, n)$$

where $z = [z_1; \cdots; z_n]$ is an auxiliary variable, and $\lambda_2 > 0$ is a tuning parameter. $(\beta_i, \beta_j) \in E_s$ and $(\beta_p, \beta_q) \in E_d (1 \le i, j, p, q \le n)$ are transformed to two quadratic penalties $c_2(\beta_i, \beta_j)$ and $c_3(\beta_p, \beta_q)$ as follows:

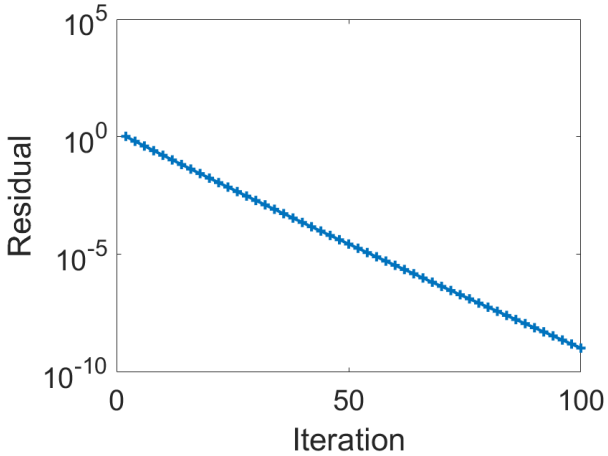
$$c_2(\beta_i,\beta_j) = \begin{cases} (\beta_i\beta_j)^2 & (\beta_i,\beta_j) \notin E_s \\ 0 & (\beta_i,\beta_j) \in E_s \end{cases}, c_3(\beta_p,\beta_q) = \begin{cases} (\beta_p\beta_q)^2 & (\beta_p,\beta_q) \notin E_d \\ 0 & (\beta_p,\beta_q) \in E_d. \end{cases}$$

Table of Contents

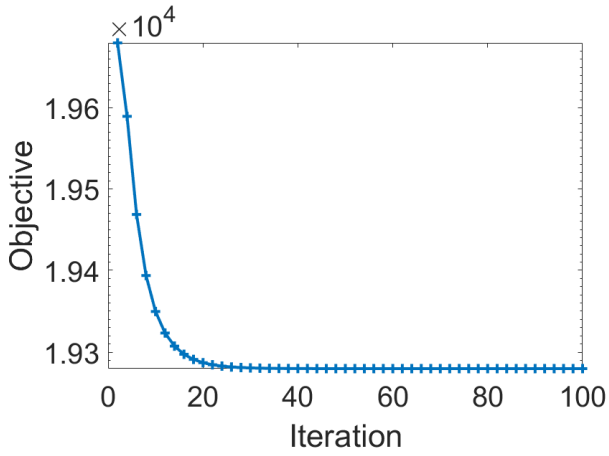
Dataset Summary

- For weak-constrained multi-task learning (i.e. Experiment I), a real-world school dataset consists of the examination scores in three years of 15,362 students from 139 secondary schools, each of which is treated as a task based on 27 input features.
- For the signed-network constrained problem (i.e. Experiment II), nine real-world datasets were used, which were obtained by randomly sampling 10% of the Twitter data from Jan 2013 to Dec 2014. Translation relationships of English, Spanish, and Portuguese were labeled as semantic links among them.

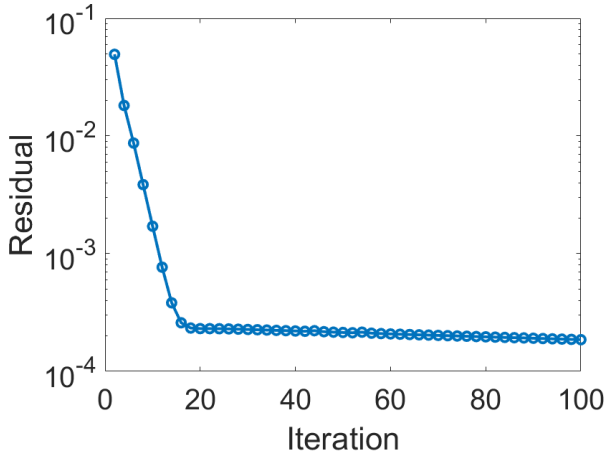
Convergence Behaviors On Experiments I and II



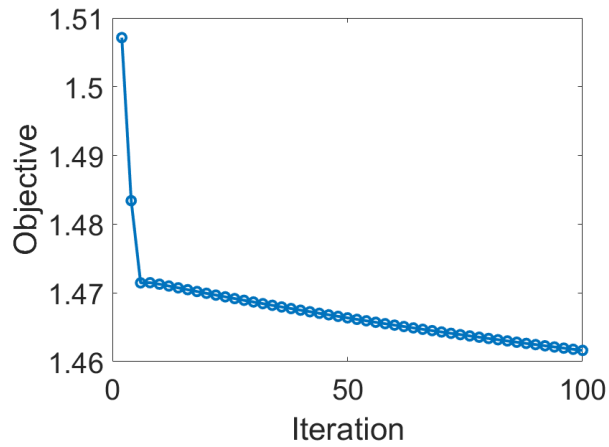
(a). Residual on Experiment I.



(b). Objective on Experiment I.



(c). Residual on the VE dataset in Experiment II.



(d). Objective on the VE dataset in Experiment II.

Figure: Convergence curves on Experiments I and II.

Performance on Experiment I

Mean					
Method	MSE	MSLE	MAE	EV	R2
JFS	114.1052	0.4531	8.4349	0.2948	0.2948
CMTL	114.9892	0.4647	8.4756	0.2876	0.2875
mtLasso	115.3143	0.4625	8.4725	0.2873	0.2873
cASO	137.8336	0.5204	9.3450	0.1606	0.1605
BCD	149.2313	0.5577	9.8057	0.1299	0.0777
ADMM($\lambda_1(1)$)	113.6975	**0.4423**	8.4024	0.2950	0.2960
ADMM($\lambda_1(2)$)	**113.2400**	0.4428	**8.3943**	**0.3002**	**0.3002**
Standard Deviation					
Method	MSE	MSLE	MAE	EV	R2
JFS	2.02	0.02	0.06	0.02	0.02
CMTL	1.85	0.02	0.05	**0.01**	**0.01**
mtLasso	1.77	0.02	0.05	**0.01**	**0.01**
cASO	7.26	0.01	0.06	**0.01**	**0.01**
BCD	1.41	0.01	0.06	0.15	**0.01**
ADMM($\lambda_1(1)$)	**0.83**	**0.005**	**0.03**	**0.01**	**0.01**
ADMM($\lambda_1(2)$)	0.95	0.01	0.04	0.02	0.02

Table: Performance in Experiment I: the proposed ADMM outperforms all comparison methods.

- $\lambda_1(1)$: $\lambda_1^k = 10^5$; $\lambda_1(2)$: $\lambda_1^{k+1} = \lambda_1^k + 10$ with $\lambda_1^k = 1$.
- MSE: Mean Squared Error; MSLE: Mean Squared Logarithmic Error; MAE: Mean Absolute Error; EV: Explained Variance.

Running time on Experiment I

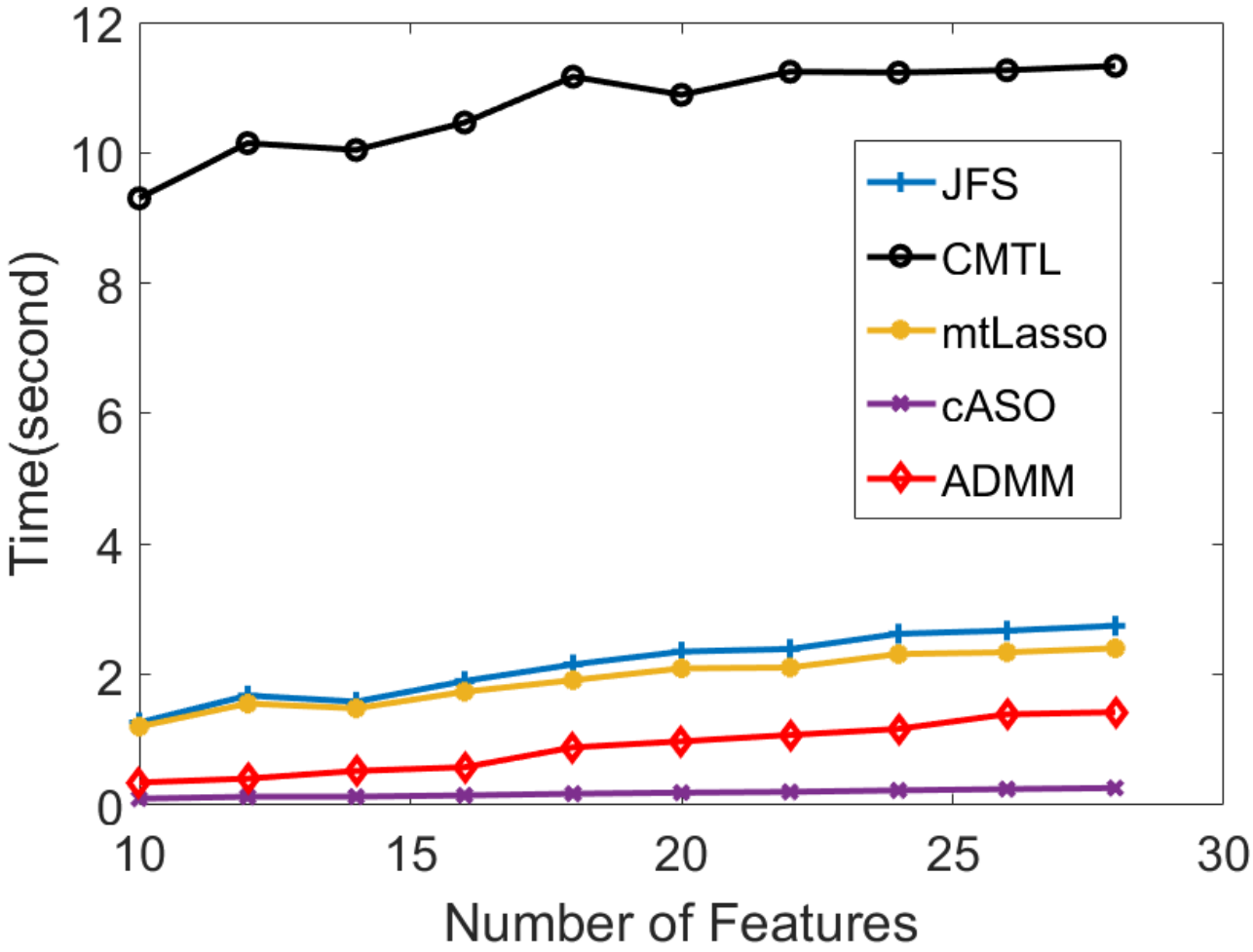


Figure: The training time of all methods in Experiment I: The training time of all methods increases linearly concerning the number of features.

Experiment on Experiment II

	BR	CL	CO	EC	EL	MX	PY	UY	VE
LogReg	0.686	0.677	0.644	0.599	0.618	0.661	0.616	0.628	0.667
LASSO	0.685	0.677	0.648	0.603	0.636	0.665	0.615	0.666	0.669
MTL	0.722	0.669	0.810	0.617	0.772	0.795	0.600	0.811	0.771
MREF	0.714	0.563	0.515	0.784	0.612	0.693	0.658	0.681	0.588
DHML	0.845	0.683	0.846	0.839	0.780	0.793	0.737	0.835	0.835
BCD	0.847	0.668	0.850	0.830	0.773	0.800	0.736	0.835	0.856
ADMM ($\lambda_2(1)$)	0.864	0.699	0.870	0.848	0.794	0.820	0.746	**0.850**	**0.867**
ADMM ($\lambda_2(2)$)	**0.867**	**0.701**	**0.872**	**0.851**	**0.798**	**0.823**	**0.747**	0.847	0.865

Table: Event forecasting performance in AUC in each of the 9 datasets.

- $\lambda_2(1)$: $\lambda_2^k = 10^5$; $\lambda_2(2)$: $\lambda_2^{k+1} = \lambda_2^k + 10$ with $\lambda_2^k = 1$.
- AUC: Area Under Receiver Operating Characteristic (ROC) Curve.

Running Time on Experiment II

	LogReg	LASSO	MTL	MREF	DHML	ADMM
BR	30193	1535	233	25889	332	**14**
CL	2981	242	35	6521	852	**11**
CO	8060	780	108	14714	87	**31**
EC	312	295	**17**	4332	46	25
EL	551	261	17	4669	33	**3**
MX	17712	2043	853	31349	175	**29**
PY	7297	527	40	9495	242	**5**
UY	748	336	20	5305	82	**3**
VE	5563	1008	49	5769	179	**28**

Table: Comparison of running time (in seconds) on 9 datasets in Experiment II: the proposed ADMM is the most efficient.

Table of Contents

Summary

In this paper, we

- propose an ADMM framework to solve the multi-convex problem.
- prove that any limit point generated by the proposed ADMM converges to a Nash point with convergence rate $o(1/k)$.
- We demonstrate two important and promising applications that are special cases of our proposed ADMM framework.
- conduct extensive experiments to validate our proposed ADMM. Experiments on ten real-world datasets demonstrate its effectiveness, scalability, and convergence properties.

Code

The code of our paper is available at
`https://github.com/xianggebenben/miADMM`.
Please contact Junxiang Wang(jwan936@emory.edu) if you have any questions.