A Fast Augmented Lagrangian Algorithm for Learning Low-Rank Matrices

Ryota Tomioka[1], Taiji Suzuki[1], Masashi Sugiyama[2], Hisashi Kashima[1]

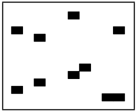
[1]The University of Tokyo
[2]Tokyo Institute of Technology

2010-06-22 @ ICML 2010

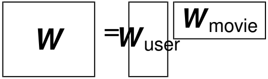
Learning low-rank matrices

1. Matrix completion [Srebro et al. 05; Abernethy et al. 09] (collaborative filtering, link prediction)

$$\boldsymbol{Y} = \boldsymbol{W}$$

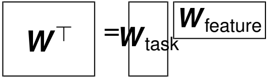


Part of $\boldsymbol{Y}$ is observed

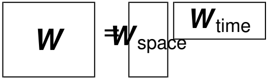


2. Multi-task learning [Argyriou et al., 07]

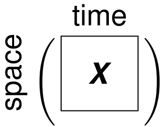
$$\boldsymbol{y} = \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b} \qquad \boldsymbol{W}^\top = \begin{bmatrix} \boldsymbol{w}_{\text{task 1}}{}^\top \\ \boldsymbol{w}_{\text{task 2}}{}^\top \\ \vdots \\ \boldsymbol{w}_{\text{task R}}{}^\top \end{bmatrix}$$



3. Predicting over matrices (classification/regression) [Tomioka & Aihara, 07]



$$y = \langle \boldsymbol{W}, \boldsymbol{X} \rangle + b$$



Problem formulation

Primal problem

$$\underset{\boldsymbol{W}\in\mathbb{R}^{R\times C}}{\text{minimize}} \quad \underbrace{f_\ell(\mathcal{A}(\boldsymbol{W})) + \phi_\lambda(\boldsymbol{W})}_{=:f(\boldsymbol{W})}$$

- f_ℓ ($\mathbb{R}^m \rightarrow \mathbb{R}$): loss function (differentiable).
- $\mathcal{A}$ ($\mathbb{R}^{R\times C} \rightarrow \mathbb{R}^m$): design matrix.
- ϕ_λ: regularizer (non-differentiable); for example, the trace norm:

$$\phi_\lambda(\boldsymbol{W}) = \lambda\|\boldsymbol{W}\|_* = \lambda\sum_{j=1}^{r}\sigma_j(\boldsymbol{W}) \quad \text{(linear sum of singular values).}$$

Separation of f_ℓ and $\mathcal{A}$ $\Rightarrow$ convergence regardless of $\mathcal{A}$ (input data).

Existing approaches and our goal

- Proximal (accelerated) gradient [Ji & Ye, 09]
  - Can keep the intermediate solution low-rank.
  - Slow for poorly conditioned design matrix $\mathcal{A}$.
  - Optimal as a first order black-box method $O(1/k^2)$.
- Interior point algorithm [Tomioka & Aihara, 07]
  - Can obtain highly precise solution in small number of iterations.
  - Only low-rank in the limit (each step can be heavy).

Can we keep the intermediate solution low-rank and still converge rapidly as an IP method?

- Dual Augmented Lagrangian (DAL) [Tomioka & Sugiyama, 09] for *sparse estimation*.
  $\Rightarrow$ M-DAL: generalized to low-rank matrix estimation.

Existing approaches and our goal

- Proximal (accelerated) gradient [Ji & Ye, 09]
  - Can keep the intermediate solution low-rank.
  - Slow for poorly conditioned design matrix $\mathcal{A}$.
  - Optimal as a first order black-box method $O(1/k^2)$.
- Interior point algorithm [Tomioka & Aihara, 07]
  - Can obtain highly precise solution in small number of iterations.
  - Only low-rank in the limit (each step can be heavy).

Can we keep the intermediate solution low-rank and still converge rapidly as an IP method?

- Dual Augmented Lagrangian (DAL) [Tomioka & Sugiyama, 09] for *sparse estimation*.
  $\Rightarrow$ M-DAL: generalized to low-rank matrix estimation.

Existing approaches and our goal

- Proximal (accelerated) gradient [Ji & Ye, 09]
  - Can keep the intermediate solution low-rank.
  - Slow for poorly conditioned design matrix $\mathcal{A}$.
  - Optimal as a first order black-box method $O(1/k^2)$.
- Interior point algorithm [Tomioka & Aihara, 07]
  - Can obtain highly precise solution in small number of iterations.
  - Only low-rank in the limit (each step can be heavy).

Can we keep the intermediate solution low-rank and still converge rapidly as an IP method?

- Dual Augmented Lagrangian (DAL) [Tomioka & Sugiyama, 09] for *sparse estimation*.
  $\Rightarrow$ M-DAL: generalized to low-rank matrix estimation.

Outline

1. Introduction
   - Why learn *low-rank* matrices?
   - Existing algorithms
2. Proposed algorithm
   - Augmented Lagrangian $\leftrightarrow$ Proximal minimization
   - Super-linear convergence
   - Generalizations
3. Experiments
   - Synthetic 10,000$\times$10,000 matrix completion.
   - BCI classification problem (learning multiple matrices).
4. Summary

M-DAL algorithm

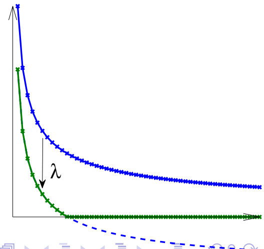
1. Initialize $\boldsymbol{W}^0$. Choose $\eta_1 \leq \eta_2 \leq \cdots$
2. Iterate until relative duality gap $< \epsilon$

$$\boldsymbol{\alpha}^t := \underset{\boldsymbol{\alpha} \in \mathbb{R}^m}{\operatorname{argmin}} \, \varphi_t(\boldsymbol{\alpha}) \quad \text{(inner objective)}$$

$$\boldsymbol{W}^{t+1} = \mathrm{ST}_{\lambda\eta_t}\left(\boldsymbol{W}^t + \eta_t \mathcal{A}^\top(\boldsymbol{\alpha}^t)\right)$$

Choosing $\boldsymbol{\alpha} = -\nabla f_\ell^t$ yields the proximal gradient (forward-backward) method.

$$\mathrm{ST}_\lambda(\boldsymbol{W}) := \underset{\boldsymbol{X} \in \mathbb{R}^{R \times C}}{\operatorname{argmin}} \left(\phi_\lambda(\boldsymbol{X}) + \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{W}\|_{\mathrm{fro}}^2\right).$$



For the trace-norm $\phi_\lambda(\boldsymbol{W}) = \lambda \|\boldsymbol{W}\|_*$,

$$\mathrm{ST}_\lambda(\boldsymbol{W}) = \boldsymbol{U} \max(\boldsymbol{S} - \lambda \boldsymbol{I}, 0) \boldsymbol{V}^\top,$$

where $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ (singular-value decomposition).

Proximal minimization [Rockafellar 76a, 76b]

1. Initialize $\boldsymbol{W}^0$.
2. Iterate:

$$\boldsymbol{W}^{t+1} = \operatorname*{argmin}_{\boldsymbol{W}} \left(f(\boldsymbol{W}) + \overbrace{\frac{1}{2\eta_t}\|\boldsymbol{W} - \boldsymbol{W}^t\|^2}^{\text{proximal term}} \right)$$

$$= \operatorname*{argmin}_{\boldsymbol{W}} \left(f_\ell(\mathcal{A}(\boldsymbol{W})) + \underbrace{\phi_\lambda(\boldsymbol{W}) + \frac{1}{2\eta_t}\|\boldsymbol{W} - \boldsymbol{W}^t\|^2}_{=: \frac{1}{\eta_t}\Phi_{\lambda\eta_t}(\boldsymbol{W};\boldsymbol{W}^t)} \right)$$

Fenchel dual (see Rockafellar 1970):

$$\min_{\boldsymbol{W}} \left(f_\ell(\mathcal{A}(\boldsymbol{W})) + \frac{1}{\eta_t}\Phi_{\lambda\eta_t}(\boldsymbol{W};\boldsymbol{W}^t) \right) = \max_{\boldsymbol{\alpha}} \left(-f_\ell^*(-\boldsymbol{\alpha}) - \frac{1}{\eta_t}\Phi^*_{\lambda\eta_t}(\eta_t\mathcal{A}^\top(\boldsymbol{\alpha});\boldsymbol{W}^t) \right)$$

$$=: -\min_{\boldsymbol{\alpha}} \varphi_t(\boldsymbol{\alpha})$$

Proximal minimization [Rockafellar 76a, 76b]

1. Initialize $\boldsymbol{W}^0$.
2. Iterate:

$$\boldsymbol{W}^{t+1} = \underset{\boldsymbol{W}}{\mathrm{argmin}}\Bigl(f(\boldsymbol{W}) + \overbrace{\frac{1}{2\eta_t}\|\boldsymbol{W} - \boldsymbol{W}^t\|^2}^{\text{proximal term}}\Bigr)$$

$$= \underset{\boldsymbol{W}}{\mathrm{argmin}}\Bigl(f_\ell(\mathcal{A}(\boldsymbol{W})) + \underbrace{\phi_\lambda(\boldsymbol{W}) + \frac{1}{2\eta_t}\|\boldsymbol{W} - \boldsymbol{W}^t\|^2}_{=: \frac{1}{\eta_t}\Phi_{\lambda\eta_t}(\boldsymbol{W};\boldsymbol{W}^t)}\Bigr)$$

Fenchel dual (see Rockafellar 1970):

$$\min_{\boldsymbol{W}}\Bigl(f_\ell(\mathcal{A}(\boldsymbol{W})) + \frac{1}{\eta_t}\Phi_{\lambda\eta_t}(\boldsymbol{W};\boldsymbol{W}^t)\Bigr) = \max_{\boldsymbol{\alpha}}\Bigl(-f_\ell^*(-\boldsymbol{\alpha}) - \frac{1}{\eta_t}\Phi^*_{\lambda\eta_t}(\eta_t\mathcal{A}^\top(\boldsymbol{\alpha});\boldsymbol{W}^t)\Bigr)$$

$$=: -\min_{\boldsymbol{\alpha}} \varphi_t(\boldsymbol{\alpha})$$

Proximal minimization [Rockafellar 76a, 76b]

1. Initialize $\boldsymbol{W}^0$.
2. Iterate:

$$\boldsymbol{W}^{t+1} = \operatorname*{argmin}_{\boldsymbol{W}} \left(f(\boldsymbol{W}) + \overbrace{\frac{1}{2\eta_t}\|\boldsymbol{W} - \boldsymbol{W}^t\|^2}^{\text{proximal term}} \right)$$

$$= \operatorname*{argmin}_{\boldsymbol{W}} \left(f_\ell(\mathcal{A}(\boldsymbol{W})) + \underbrace{\phi_\lambda(\boldsymbol{W}) + \frac{1}{2\eta_t}\|\boldsymbol{W} - \boldsymbol{W}^t\|^2}_{=: \frac{1}{\eta_t}\Phi_{\lambda\eta_t}(\boldsymbol{W};\boldsymbol{W}^t)} \right)$$

Fenchel dual (see Rockafellar 1970):

$$\min_{\boldsymbol{W}} \left(f_\ell(\mathcal{A}(\boldsymbol{W})) + \frac{1}{\eta_t}\Phi_{\lambda\eta_t}(\boldsymbol{W};\boldsymbol{W}^t) \right) = \max_{\boldsymbol{\alpha}} \left(-f_\ell^*(-\boldsymbol{\alpha}) - \frac{1}{\eta_t}\Phi^*_{\lambda\eta_t}(\eta_t\mathcal{A}^\top(\boldsymbol{\alpha});\boldsymbol{W}^t) \right)$$

$$=: -\min_{\boldsymbol{\alpha}} \varphi_t(\boldsymbol{\alpha})$$

Definition

- $\boldsymbol{W}^*$: the unique minimizer of the objective $f(\boldsymbol{W})$.
- $\boldsymbol{W}^t$: sequence generated by the M-DAL algorithm with

$$\|\nabla\varphi_t(\boldsymbol{\alpha}^t)\| \leq \sqrt{\frac{\gamma}{\eta_t}}\|\boldsymbol{W}^{t+1} - \boldsymbol{W}^t\|_{\text{fro}}$$

($1/\gamma$: Lipschitz constant of ∇f_ℓ.)

Assumption

There is a constant $\sigma > 0$ such that

$$f(\boldsymbol{W}^{t+1}) - f(\boldsymbol{W}^*) \geq \sigma\|\boldsymbol{W}^{t+1} - \boldsymbol{W}^*\|_{\text{fro}}^2 \quad (t = 0, 1, 2, \ldots).$$

Theorem: Super-linear convergence

$$\|\boldsymbol{W}^{t+1} - \boldsymbol{W}^*\|_{\text{fro}} \leq \frac{1}{\sqrt{1 + 2\sigma\eta_t}}\|\boldsymbol{W}^t - \boldsymbol{W}^*\|_{\text{fro}}.$$

I.e., $\boldsymbol{W}^t$ converges super-linearly to $\boldsymbol{W}^*$ if η_t is increasing.

Why is M-DAL efficient?

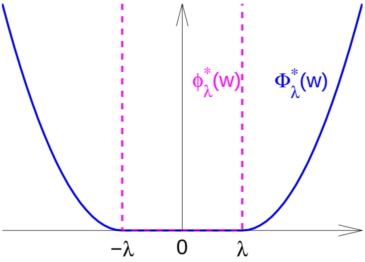
(1) Proximation wrt ϕ_λ is analytic (though non-smooth):

$$\boldsymbol{W}^{t+1} = \mathrm{ST}_{\eta_t\lambda}\left(\boldsymbol{w}^t + \eta_t \mathcal{A}^\top(\boldsymbol{\alpha}^t)\right)$$

(2) Inner minimization is smooth:

$$\boldsymbol{\alpha}^t = \underset{\boldsymbol{\alpha}\in\mathbb{R}^m}{\mathrm{argmin}}\Bigg(\underbrace{f_\ell^*(-\boldsymbol{\alpha})}_{\text{Differentiable}} + \frac{1}{2\eta_t} \underbrace{\|\mathrm{ST}_{\eta_t\lambda}(\boldsymbol{W}^t + \eta_t\mathcal{A}^\top(\boldsymbol{\alpha}))\|_{\mathrm{fro}}^2}_{=\Phi_\lambda^*(\cdot)} \Bigg)$$

(linear to the estimated rank)



Generalizations

1. Learning multiple matrices

$$\phi_\lambda(\boldsymbol{W}) = \lambda \sum_{k=1}^{K} \|\boldsymbol{W}^{(k)}\|_* = \lambda \left\| \begin{pmatrix} \boldsymbol{W}^{(1)} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \boldsymbol{W}^{(K)} \end{pmatrix} \right\|_*$$

   - No need to form the big matrix.
   - Can be used to select informative data sources and learn feature extractors simultaneously.

2. General spectral regularization

$$\phi_\lambda(\boldsymbol{W}) = \sum_{j=1}^{r} g_\lambda(\sigma_j(\boldsymbol{W}))$$

for any convex function g_λ for which the proximal operator:

$$\mathrm{ST}_\lambda^g(\sigma_j) = \underset{x\in\mathbb{R}}{\mathrm{argmin}} \left(g_\lambda(x) + \frac{1}{2}(x - \sigma_j)^2 \right)$$

can be computed in closed form.

Synthetic experiment 1: low-rank matrix completion

Large scale & structured.

- True matrix $\boldsymbol{W}^*$: $10,000 \times 10,000$ (100M elements), low rank.
- Observation: randomly chosen m elements (sparse).

$\Rightarrow$ Quasi-Newton method for the minimization of $\varphi_t(\boldsymbol{\alpha})$

$\Rightarrow$ No need to form the full matrix!

m=1,200,000 m=2,400,000

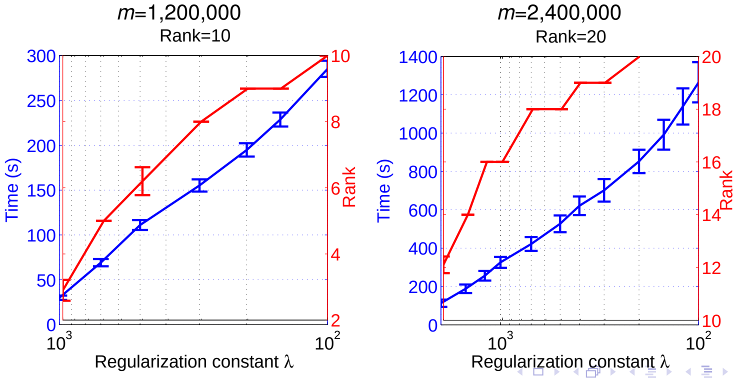
Synthetic experiment 1: low-rank matrix completion

Large scale & structured.

- True matrix $\boldsymbol{W}^*$: $10,000 \times 10,000$ (100M elements), low rank.
- Observation: randomly chosen m elements (sparse).

$\Rightarrow$ Quasi-Newton method for the minimization of $\varphi_t(\boldsymbol{\alpha})$

$\Rightarrow$ No need to form the full matrix!



Rank=10

Rank=10, #observations m=1,200,000

λ	time (s)	#outer	#inner	rank	S-RMSE
1000	33.1 ($\pm$2.0)	5 ($\pm$0)	8 ($\pm$0)	2.8 ($\pm$0.4)	0.0158 ($\pm$0.0024)
700	77.1 ($\pm$5.6)	11 ($\pm$0)	18 ($\pm$0)	5 ($\pm$0)	0.0133 ($\pm$0.0008)
500	124 ($\pm$7.2)	17 ($\pm$0)	28 ($\pm$0)	6.4 ($\pm$0.5)	0.0113 ($\pm$0.0015)
300	174 ($\pm$8.0)	23 ($\pm$0)	38.4 ($\pm$0.84)	8 ($\pm$0)	0.00852 ($\pm$0.00039)
200	220 ($\pm$9.9)	29 ($\pm$0)	48.4 ($\pm$0.84)	9 ($\pm$0)	0.00767 ($\pm$0.00031)
150	257 ($\pm$9.9)	35 ($\pm$0)	58.4 ($\pm$0.84)	9 ($\pm$0)	0.00498 ($\pm$0.00026)
100	319 ($\pm$11)	41 ($\pm$0)	70 ($\pm$0.82)	10 ($\pm$0)	0.00743 ($\pm$0.00013)

- #inner iterations is roughly 2 times #outer iterations.
  $\Leftarrow$ Because we don't need to solve the inner problem very precisely!

Rank=20

Rank=20, #observations m=2,400,000

λ	time (s)	#outer	#inner	rank	S-RMSE
2000	112 ($\pm$19)	6 ($\pm$0)	15.1 ($\pm$1.0)	12.1 ($\pm$0.3)	0.011 ($\pm$0.002)
1500	188 ($\pm$22)	11 ($\pm$0)	24.1 ($\pm$1.0)	14 ($\pm$0)	0.0094 ($\pm$0.001)
1200	256 ($\pm$25)	15 ($\pm$0)	31.1 ($\pm$1.0)	16 ($\pm$0)	0.0090 ($\pm$0.0008)
1000	326 ($\pm$29)	19 ($\pm$0)	38.1 ($\pm$1.0)	16 ($\pm$0)	0.0073 ($\pm$0.0007)
700	421 ($\pm$36)	24 ($\pm$0)	48.1 ($\pm$1.0)	18 ($\pm$0)	0.0065 ($\pm$0.0004)
500	527 ($\pm$44)	29 ($\pm$0)	57.1 ($\pm$1.0)	18 ($\pm$0)	0.0042 ($\pm$0.0003)
400	621 ($\pm$48)	34 ($\pm$0)	66.1 ($\pm$1.0)	19 ($\pm$0)	0.0044 ($\pm$0.0002)
300	702 ($\pm$59)	38.5 ($\pm$0.5)	74.1 ($\pm$1.5)	19 ($\pm$0)	0.0030 ($\pm$0.0003)
200	852 ($\pm$61)	43.6 ($\pm$0.5)	83.9 ($\pm$2.3)	20 ($\pm$0)	0.0039 ($\pm$0.0001)
150	992 ($\pm$78)	48.4 ($\pm$0.7)	92.5 ($\pm$1.5)	20 ($\pm$0)	0.0024 ($\pm$0.0002)
120	1139 ($\pm$94)	53.4 ($\pm$0.7)	102 ($\pm$1.5)	20 ($\pm$0)	0.0016 ($\pm 6\times10^{-5}$)
100	1265 ($\pm$105)	57.7 ($\pm$0.8)	109 ($\pm$2.4)	20 ($\pm$0)	0.0013 ($\pm 8\times10^{-5}$)

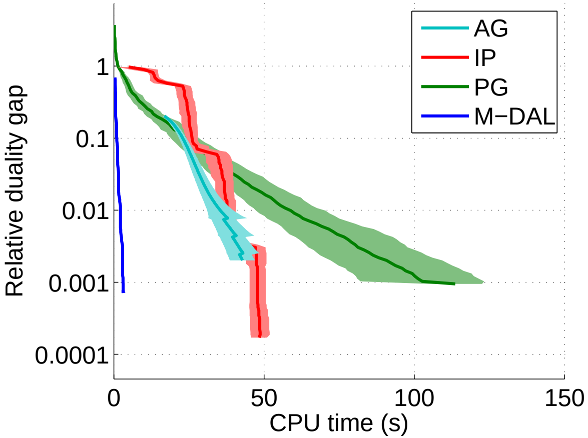
Synthetic experiment 2: classifying matrices

- True matrix $\boldsymbol{W}^*$: 64×64, rank=16.
- Classification problem (# samples $m = 1000$):

$$f_\ell(\boldsymbol{z}) = \sum_{i=1}^{m} \log(1 + \exp(-y_i z_i))$$

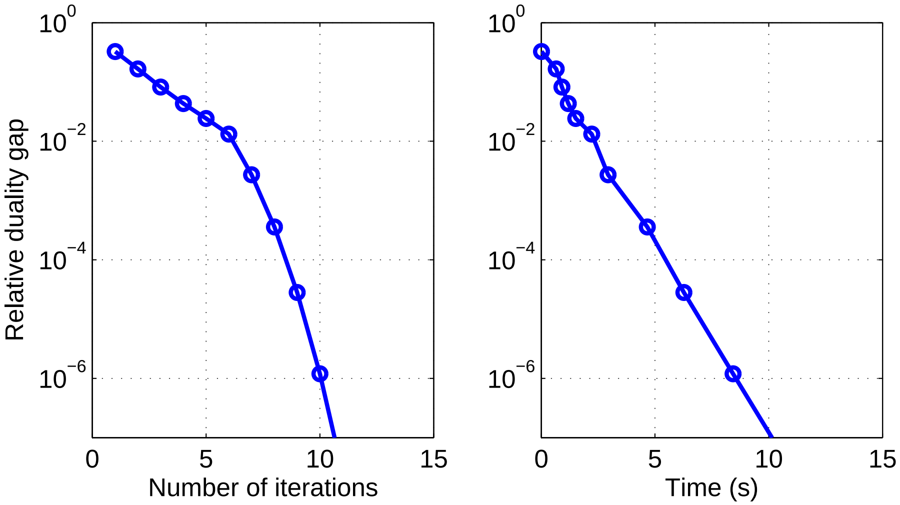
- Design matrix $\mathcal{A}$: 1000×64^2 (dense) each example drawn from Wishart distribution.
- $\lambda = 800$ (chosen to roughly reproduce rank=16).
- Medium scale & dense $\Rightarrow$ Newton method for minimizing $\varphi_t(\boldsymbol{\alpha})$ (works better when the condition is poor).
- Methods:
  - M-DAL (proposed)
  - IP (interior point method [T& A 2007])
  - PG (projected gradient method [T& S 2008])
  - AG (accelerated gradient method [Ji & Ye 2009])

Comparison



- M-DAL (proposed)
- IP (interior point method [T& A 2007])
- PG (projected gradient method [T& S 2008])
- AG (accelerated gradient method [Ji & Ye 2009])

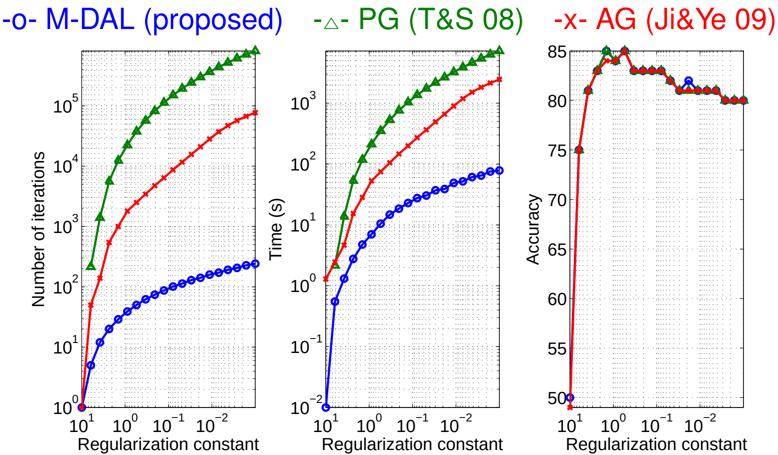
Super-linear convergence



Benchmark experiment: BCI dataset

- BCI competition 2003 dataset IV.
- Task: predict the upcoming finger movement is right or left.
- Original data: multichannel EEG time-series 28 channels $\times$ 50 time-points (500ms long)
- Each training example is preprocessed into three matrices and whitened [T&M 2010]:
  - First order (<20Hz) component: 28$\times$50 matrix.
  - Second order (alpha 7-15Hz) component: 28$\times$28 (covariance)
  - Second order (beta 15-30Hz) component: 28$\times$28 (covariance)
- 316 training examples. 100 test samples.

BCI dataset (regularization path)



- Stopping criterion: RDG$\leq 10^{-3}$.
- Note: these are costs for obtaining the entire regularization path.

Summary

- M-DAL: Dual Augmented Lagrangian algorithm for learning low-rank matrices.
- Theoretically shown that M-DAL converges superlinearly
  $\Rightarrow$requires small number of steps.
- Separable regularizer + differentiable loss function
  $\Rightarrow$each step can be efficiently solved.
- Empirically we can solve matrix completion problem of size 10,000$\times$10,000 in roughly 5min.
- Training a low-rank classifier that automatically combines multiple data sources (matrices) is sped up by a factor of 10–100.

Future work:

- Other AL algorithms.
- Large scale and unstructured (or badly conditioned) problems.

Background

- Convex conjugate of f

$$f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}\in\mathbb{R}^n} \left(\langle \boldsymbol{y}, \boldsymbol{x}\rangle - f(\boldsymbol{x})\right)$$

- Convex conjugate of a sum:

$$(f+g)^*(\boldsymbol{y}) = (f^* \oplus g^*)(\boldsymbol{\alpha}) = \inf_{\boldsymbol{\alpha}} \left(f^*(\boldsymbol{\alpha}) + g^*(\boldsymbol{y}-\boldsymbol{\alpha})\right)$$

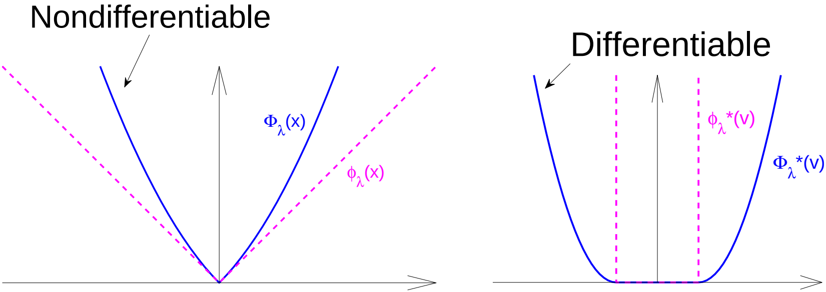
- Fenchel dual:

$$\inf_{\boldsymbol{x}\in\mathbb{R}^n} \left(f(\boldsymbol{x}) + g(\boldsymbol{x})\right) = \sup_{\boldsymbol{\alpha}\in\mathbb{R}^n} \left(-f^*(\boldsymbol{\alpha}) - g^*(-\boldsymbol{\alpha})\right)$$

Inf-convolution

Inf-convolution (envelope function):

$$\begin{aligned}\Phi_\lambda^*(\boldsymbol{V};\boldsymbol{W}^t) &= (\phi_\lambda + \frac{1}{2}\|\cdot - \boldsymbol{W}^t\|)^*(\boldsymbol{V})\\ &= \inf_{\boldsymbol{Y}\in\mathbb{R}^{R\times C}}\left(\phi_\lambda^*(\boldsymbol{Y}) + \frac{1}{2}\|\boldsymbol{W}^t + \boldsymbol{V} - \boldsymbol{Y}\|^2\right)\\ &= \Phi_\lambda^*(\boldsymbol{W}^t + \boldsymbol{V};\boldsymbol{0}) = \Phi_\lambda^*(\boldsymbol{W}^t + \boldsymbol{V})\end{aligned}$$



M-DAL algorithm for the trace-norm

$$\boldsymbol{W}^{t+1} = \mathrm{ST}_{\lambda\eta_t}\left(\boldsymbol{W}^t + \eta_t \mathcal{A}^\top(\boldsymbol{\alpha}^t)\right)$$

where

$$\boldsymbol{\alpha}^t = \operatorname*{argmin}_{\boldsymbol{\alpha}} \underbrace{\left(f_\ell^*(-\boldsymbol{\alpha}) + \overbrace{\frac{1}{2\eta_t}\|\mathrm{ST}_{\lambda\eta_t}(\boldsymbol{W}^t + \eta_t\mathcal{A}^\top(\boldsymbol{\alpha}))\|^2}^{\frac{1}{\eta_t}\Phi^*_{\lambda\eta_t}(\boldsymbol{W}^t+\eta_t\mathcal{A}^\top(\boldsymbol{\alpha}))} \right)}_{\varphi_t(\boldsymbol{\alpha})}$$

- $\varphi_t(\boldsymbol{\alpha})$ is differentiable:

$$\nabla\varphi_t(\boldsymbol{\alpha}) = -\nabla f_\ell^*(-\boldsymbol{\alpha}) + \mathcal{A}(\mathrm{ST}_{\lambda\eta_t}(\boldsymbol{W}^t + \eta_t\mathcal{A}^\top(\boldsymbol{\alpha})))$$

$\nabla^2\varphi_t(\boldsymbol{\alpha})$: a bit more involved but can be computed (Wright 92)

Minimizing the inner objective $\varphi_t(\boldsymbol{\alpha})$

- $\varphi_t(\boldsymbol{\alpha})$, $\nabla\varphi_t(\boldsymbol{\alpha})$ can be computed using only the singular values $\sigma_j(\boldsymbol{W}^t_{\boldsymbol{\alpha}}) \geq \lambda\eta_t$ (and the corresponding SVs).
  $\Rightarrow$ no need to compute full SVD of $\boldsymbol{W}^t_{\boldsymbol{\alpha}} := \boldsymbol{W}^t + \eta_t\mathcal{A}^\top(\boldsymbol{\alpha})$.
- But, computation of $\nabla^2\varphi_t(\boldsymbol{\alpha})$ requires full SVD.

Consequence:

- When $\mathcal{A}^\top(\boldsymbol{\alpha})$ is structured, computing the above SVD is cheap.
  E.g., matrix completion: $\mathcal{A}^\top(\boldsymbol{\alpha})$ is sparse with only m non-zeros.

$$\boxed{\boldsymbol{W}^t_{\boldsymbol{\alpha}}} = \boxed{\boldsymbol{W}^t_L}\,\boxed{\boldsymbol{W}^t_R} + \eta_t \boxed{}$$

  $\Rightarrow$ Quasi-Newton method (maintains factorized $\boldsymbol{W}^t$ and scalable)
- When $\mathcal{A}^\top(\boldsymbol{\alpha})$ is *not* structured,
  $\Rightarrow$ Full Newton method (converges faster and more stable)

Primal Proximal Minimization A (M-DAL)

$$\boldsymbol{W}^{t+1} = \underset{\boldsymbol{W}}{\mathrm{argmin}} \left(f_\ell(\mathcal{A}(\boldsymbol{W})) + \left(\phi_\lambda(\boldsymbol{W}) + \frac{1}{2\eta_t} \|\boldsymbol{W} - \boldsymbol{W}^t\|_{\mathrm{fro}}^2 \right) \right)$$

- Primal Proximal Minimization B

$$\boldsymbol{W}^{t+1} = \underset{\boldsymbol{W}}{\mathrm{argmin}} \left(\phi_\lambda(\boldsymbol{W}) + \left(f_\ell(\mathcal{A}(\boldsymbol{W})) + \frac{1}{2\eta_t} \|\boldsymbol{W} - \boldsymbol{W}^t\|_{\mathrm{fro}}^2 \right) \right)$$

- Dual Proximal Minimization A ($\sim$ split Bregman iteration)

$$\boldsymbol{\alpha}^{t+1} = \underset{\boldsymbol{\alpha}}{\mathrm{argmin}} \left(f_\ell^*(-\boldsymbol{\alpha}) + \left(\phi_\lambda^*(\mathcal{A}^\top(\boldsymbol{\alpha})) + \frac{1}{2\eta_t} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}^t\|^2 \right) \right)$$

- Dual Proximal Minimization B

$$\boldsymbol{\alpha}^{t+1} = \underset{\boldsymbol{\alpha}}{\mathrm{argmin}} \left(\phi_\lambda^*(\mathcal{A}^\top(\boldsymbol{\alpha})) + \left(f_\ell^*(-\boldsymbol{\alpha}) + \frac{1}{2\eta_t} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}^t\|^2 \right) \right)$$

References

- Abernethy et al. A new approach to collaborative filtering: Operator estimation with spectral regularization. JMLR 10, 2009.
- Argyriou et al. Multi-task feature learning In NIPS 19, 2007.
- Ji & Ye. An accelerated gradient method for trace norm minimization. In ICML 2009.
- Rockafellar. Monotone operators and the proximal point algorithm. SIAM J. on Control and Optimization 14, 1976a.
- Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. Math. of Oper. Res. 1, 1976b.
- Srebro et al. Maximum-margin matrix factorization. In NIPS 17, 2005.
- Tomioka & Aihara. Classifying matrices with a spectral regularization. In ICML 2007.
- Tomioka et al. Super-Linear Convergence of Dual Augmented-Lagrangian Algorithm for Sparsity Regularized Estimation. Arxiv:0911.4046, 2010.
- Tomioka & Sugiyama. Sparse learning with duality gap guarantee. In NIPS workshop OPT 2008 Optimization for Machine Learning, 2008.
- Tomioka & Müller. A regularized discriminative framework for EEG analysis with application to brain-computer interface. Neuroimage 49, 2010.
- Wright. Differential equations for the analytic singular value decomposition of a matrix. Numer. Math. 63, 1992.