Matrix Learning Problems and First-Order Optimization

Andreas Argyriou

Ecole Centrale Paris

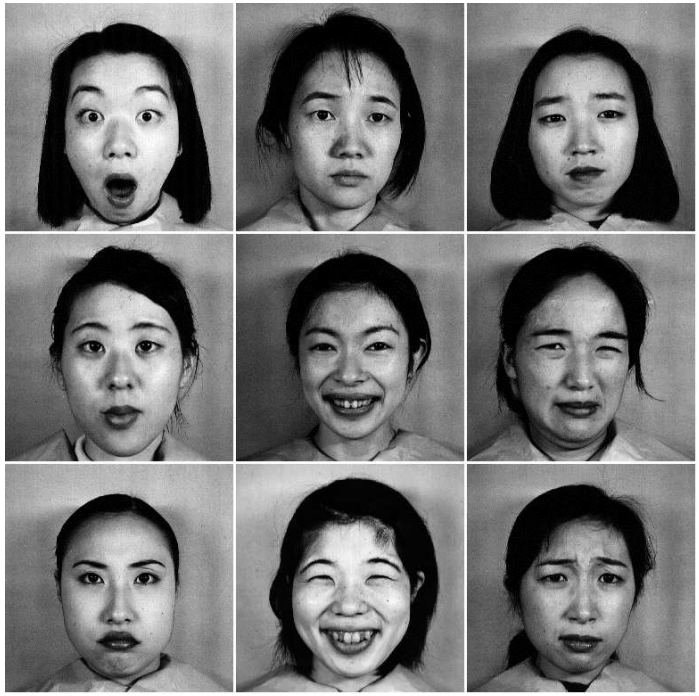
Outline

- Proximal and accelerated methods, scalability
- Smoothing methods, PRISMA
- Applications: machine learning problems with matrices
- (joint work with F. Orabona and N. Srebro at TTI Chicago)

Matrix Problems in Machine Learning

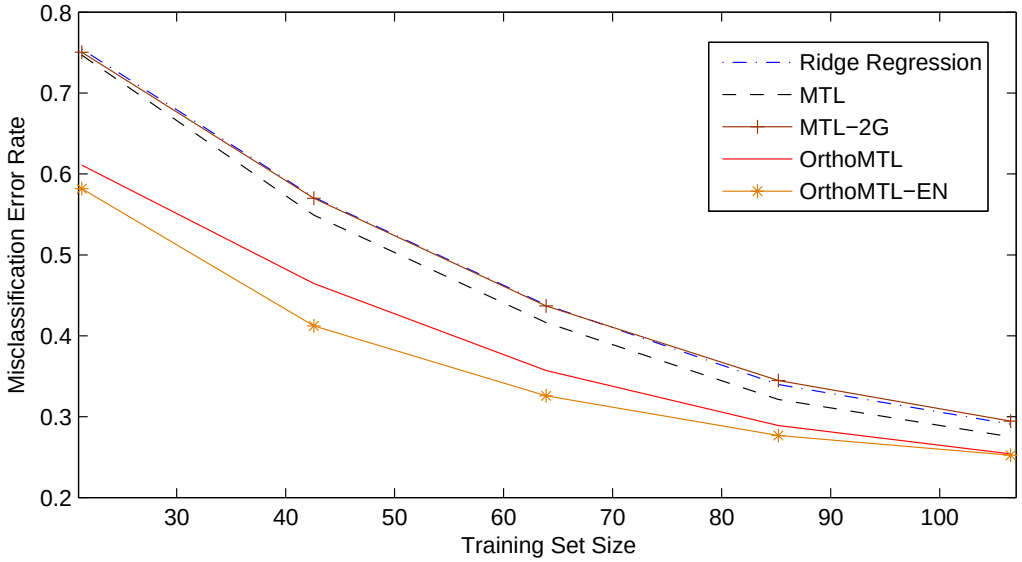
- Collaborative filtering, matrix completion, multitask learning
- Clustering
- Background extraction from surveillance video
- Gene expression networks; computer vision (multiple object detection, reconstruction with video sequences etc.)
- Interesting learning problems nowadays tend to exhibit *matrix* structure

Multitask Learning



Learning one task (learning one person's emotions) can be boosted by learning multiple *related* tasks (multiple persons' emotions) jointly

Multitask Learning



- Learning one task can be boosted by learning multiple *related* tasks jointly
- Typically it can be formulated as a matrix problem

Matrix Completion

- Marketing databases, collaborative filtering, recommendation systems (e.g. Netflix); task = product preferences for each person

Description				
Closure	Type of winery	Type of wine	Price	Your rating
Metacork	International	Blush red	$25	
Metacork	Mid-sized regional	Dry white	$20	
Traditional cork	Small boutique	Dry red	$20	
Screwcap	International	Dry red	$30	
Metacork	Small boutique	Aromatic white	$30	
Traditional cork	International	Dry white	$15	
Screwcap	Large national	Blush red	$20	
Synthetic cork	International	Aromatic white	$20	

First-Order and Proximal Optimization

- Methodology for first-order optimization with long history [Levitin & Polyak 1966; Martinet 1970]; proximity map dates back to [Moreau 1965]

- Main advantage: **scalability**; can be used when covariance/Hessian $X^\top X$ is too large to compute or store in memory

- Essentially these methods are generalizations of *gradient descent*

- In other words: linearize the smooth part f of the objective around a point α_k that moves towards a solution

Gradient Descent

- Consider the optimization problem

$$\text{minimize } f$$

under the assumptions

- $f : \mathbb{R}^d \to \mathbb{R}$ is a *convex* function
- f is *smooth* in the sense

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \qquad \forall x, y \in \mathbb{R}^d$$

i.e. its gradient is Lipschitz

Gradient Descent (contd.)

Use a simple gradient descent scheme

$$\beta_{t+1} \leftarrow \beta_t - \frac{1}{L}\nabla f(\beta_t)$$

- Essentially, we do not need to know the Hessian (unlike e.g. Newton-type methods); we just need an upper bound L on its norm
- Some nice properties
  - The iterates converge to a minimizer of f: $\beta_t \to \beta^*$ (weak convergence)
  - Convergence rate in the objective is $\mathcal{O}\left(\frac{1}{t}\right)$

$$f(x_t) - f(x^*) \leq \frac{L\|x_1 - x^*\|^2}{2t}$$

Accelerated Gradient Descent

- Proposed by Nesterov in 1983
- Achieves a better rate of $O(\frac{1}{t^2})$ in the objective value

$$f(x_t) - f(x^*) \leq \frac{2L\|x_1 - x^*\|^2}{(t+1)^2}$$

- In fact, $O(\frac{1}{t^2})$ rate is *optimal*: it can be shown that specific examples of f require at least this convergence rate
- Idea: use memory (i.e. momentum) with a carefully chosen *affine* weighting of the previous estimates

Accelerated Gradient Descent (contd.)

Initialization: $\beta_1 = \alpha_1$
for $t = 1, 2, \ldots$ **do**
  Compute $\beta_{t+1} \leftarrow \alpha_t - \frac{1}{L}\nabla f(\alpha_t)$
  $\alpha_{t+1} \leftarrow \rho_{t+1}\,\beta_{t+1} - (\rho_{t+1} - 1)\beta_t$
end for

- Sequence ρ_t should satisfy a certain recursive inequality; $\rho_t = 2 - \frac{3}{t+2}$ is one possible choice

- Several variants of this method satisfy similar bounds (e.g. infinite memory)

Forward-Backward Splitting

- Extend to the problem

$$\min\{f(\beta) + \omega(\beta) : \beta \in \mathbb{R}^d\}$$

Assumptions

(i) f, ω convex, l.s.c., $\omega : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$
(ii) ∇f Lipschitz continuous with constant L
(iii) a minimizer exists

E.g. $f(\beta) = \frac{1}{2}\|X\beta - y\|_2^2$, $L = \|X\|^2$, $\omega = \|\cdot\|_1$

Usually ω is *nonsmooth*;
it can also be nonfinite, e.g. an indicator of a closed convex constraint
(0 inside the feasible set, $+\infty$ outside)

Forward-Backward Splitting (contd.)

Input: $\beta_1 \in \mathbb{R}^n$
for $t = 1, 2, \ldots$ **do**
(a) $\alpha_{t+1} \leftarrow \beta_t - \frac{1}{L}\nabla f(\beta_t)$
(b) $\beta_{t+1} \leftarrow \mathrm{prox}_{\frac{1}{L}\omega}(\alpha_{t+1})$
end for

The *proximity map* [Moreau 1965] is a generalized projection

$$\mathrm{prox}_\omega(x) := \operatorname*{argmin}_y \left\{ \frac{1}{2}\|y - x\|_2^2 + \omega(y) \right\}$$

- Note: the minimizer is unique since $\|\cdot\|^2$ is strongly convex
- In the case $\omega = \delta_C$, the indicator of a closed convex set C, $\mathrm{prox}_\omega = \mathrm{Proj}_C$

Interpretations

Alternating proximity maps

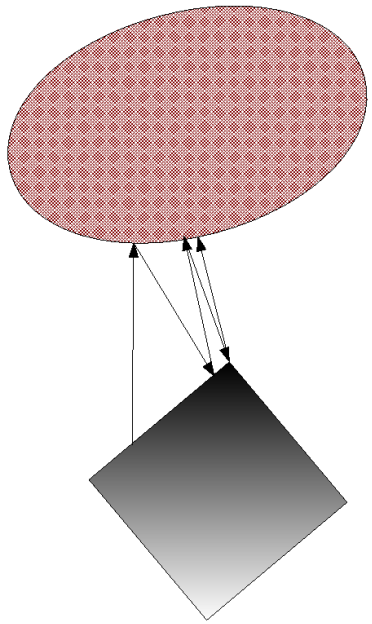
Let $h := f^* - \frac{1}{2L}\|\cdot\|^2$

Input: $\beta_1 \in \mathbb{R}^n$
for $t = 1, 2, \ldots$ **do**
(a) $\alpha_{t+1} \leftarrow \operatorname{prox}_{\frac{1}{L}h^*}(\beta_t)$
(b) $\beta_{t+1} \leftarrow \operatorname{prox}_{\frac{1}{L}\omega}(\alpha_{t+1})$
end for

- I.e. $\frac{1}{L}\nabla f = I - \operatorname{prox}_{\frac{1}{L}h^*}$
- f convex, $\frac{1}{L}\nabla f$ nonexpansive $\implies$ $\frac{1}{L}\nabla f$ firmly nonexpansive [Baillon-Haddad 1977]

Interpretations

- In other words, forward-backward splitting extends alternating projections (which date at least to the 1930s)



Forward-Backward Splitting (Rate)

- Forward-backward algorithm subsumes several methods (projected gradient, alternating projections, projected Landweber)

- Similar rate to that of gradient descent

$$f(x_t) + \omega(x_t) - (f(x^*) + \omega(x^*)) \leq \frac{L\|x_1 - x^*\|^2}{2t}$$

Accelerated Proximal Methods

- Proposed in various papers of Nesterov; also studied by Tseng, Beck & Teboulle etc.

- Similar trick as in accelerated gradient descent

Initialization: $\beta_1 = \alpha_1$
for $t = 1, 2, \ldots$ **do**
  Compute $\beta_{t+1} \leftarrow \mathrm{prox}_{\frac{\omega}{L}}\left(\alpha_t - \frac{1}{L}\nabla f(\alpha_t)\right)$
  $\alpha_{t+1} \leftarrow \rho_{t+1}\,\beta_{t+1} - (\rho_{t+1} - 1)\beta_t$
end for

$$f(x_t) + \omega(x_t) - (f(x^*) + \omega(x^*)) \leq \frac{2L\|x_1 - x^*\|^2}{(t+1)^2}$$

Specific Applications

- In recent years these methods have been applied extensively in machine learning, compressed sensing, signal processing etc.
  see [Beck & Teboulle],[Becker, Candes & Grant],[Cai,Candes& Shen],[Daubechies, Defrise & De Mol],[Duchi & Singer], [Jenatton et al.],[Liu & Ye],[Mosci et al.],[Zhang, Saha & Vishwanathan] etc.

- Fast rate depends on **exact** computation of the proximity map,
  i.e. *exact* minimization of $\frac{L}{2}\|u-x\|_2^2+\omega(u)$ wrt. u in a finite number of steps

- Rate is preserved with inexact prox computation if latter is linearly convergent [Devolder et al., Schmidt et al.]

Specific Applications

- Exact computation applies to many cases of interest
  - Iterative soft thresholding $\omega = \ell_1$
  - Regularizations, constraints on ℓ_∞
  - Structured sparsity, group Lasso $\ell_{2,1}$
  - Singular value thresholding (trace norm)
  - Fused Lasso, total variation etc.

A More General Class of Problems

$$\min\{f(\beta) + g(\beta) + h(\beta) : \beta \in \mathbb{R}^d\}$$

where f is smooth (has Lipschitz gradient) but g, h may be nonsmooth

- Known convergence rates do not apply, since the operator prox_{g+h} cannot be computed exactly (requires a numerical algorithm)

- Our approach: introduce a helpful additional assumption i.e. g is Lipschitz

$$|g(x) - g(y)| \leq \rho_g \|x - y\|$$

Nesterov's Smoothing Methods

- They solve
$$\min\{f(\beta) + g(\beta) : \beta \in C\}$$
where f is smooth and g is Lipschitz continuous

- Involve prox_g and two projections onto the domain at each iteration

- $\mathcal{O}(\frac{1}{t})$ convergence guarantee

- Algorithm and convergence guarantee rely on *the domain being bounded*

- Our goal: remove boundedness assumption; express convergence rate in terms of ρ_g instead

Smoothing of Nonsmooth Functions

- Main idea: smooth the nonsmooth Lipschitz part

  Let $\varphi : \mathbb{R}^d \to \mathbb{R}$ be a convex, ρ-Lipschitz function, then

$$\varphi_\mu \leq \varphi \leq \varphi_\mu + \frac{1}{2}\mu\rho^2$$

where φ_μ is a smoothing, the *Moreau envelope*, of φ

$$\varphi_\mu(x) := \inf\left\{\frac{1}{2\mu}\|x-u\|^2 + \varphi(u) : u \in \mathbb{R}^n\right\}$$

PRISMA

PRoximal Iterative SMoothing Algorithm

Parameters $\{\mu_t > 0 : t \in \mathbb{N}\}$, $\{\theta_t > 0 : t \in \mathbb{N}\}$
Initialize $\beta_1 = y_1$
for $t = 1, 2, \ldots$ **do**
$L_t \leftarrow L_f + \frac{1}{\mu_t}$
$\beta_{t+1} \leftarrow \mathrm{prox}_{\frac{1}{L_t}h}\left((1 - \frac{1}{L_t\mu_t})y_t - \frac{1}{L_t}\nabla f(y_t) + \frac{1}{L_t\mu_t}\mathrm{prox}_g(y_t, \mu_t)\right)$
$y_{t+1} \leftarrow \beta_{t+1} + \theta_{t+1}\left(\frac{1}{\theta_t} - 1\right)(\beta_{t+1} - \beta_t)$
end for

Set $\theta_{t+1} = \frac{1}{2} + \frac{1}{2}\sqrt{1 + \frac{4\theta_t^2 L_{t+1}}{L_t}}$, $\mu_t = \frac{1}{at}$, $a > 0$

PRISMA (convergence rate)

Theorem

$$F(\beta_{t+1}) - F(\hat{\beta}) \leq 2\frac{L_f + at}{(t+1)^2}\Big[\|\hat{\beta} - \beta_1\|^2 + \frac{\rho_g^2}{a}\Big(\frac{1}{L_f + a} + \frac{3}{2a}\log\frac{L_f + at}{L_f + a}\Big)\Big]$$

- Rate is $\mathcal{O}\left(\frac{\log t}{t}\right)$
- Extended applicability comes at a cost compared to accelerated Nesterov method; additional log factor compared to Nesterov smoothing

Applications: Matrix Completion

- *Max norm* has been used for matrix completion and clustering [Srebro et al.], [Jalali & Srebro]

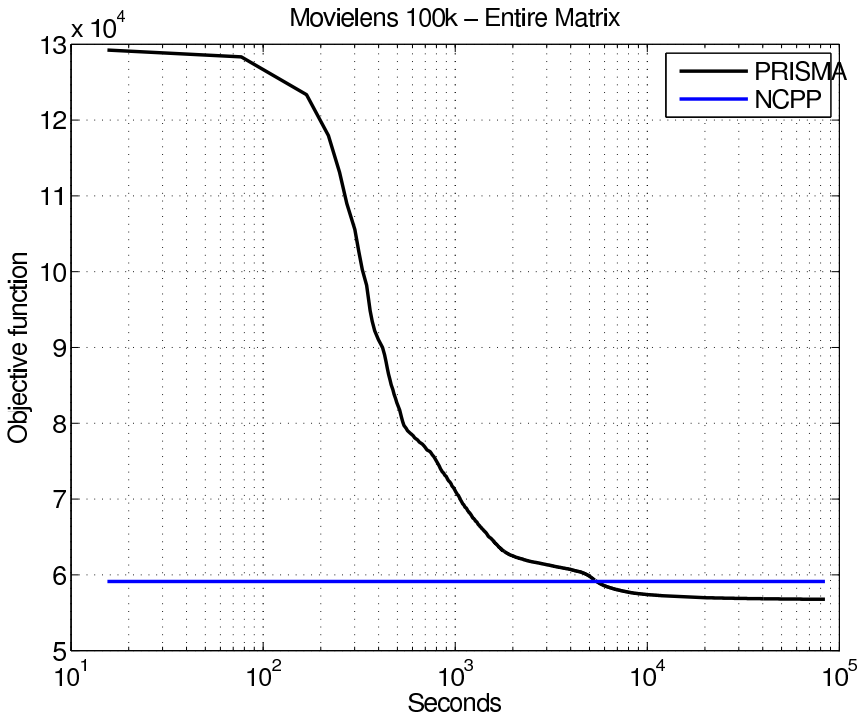
- SDP representation

$$\|W\|_{max} = \min t \quad \text{s.t.} \quad \left(\begin{smallmatrix} A & X \\ X^\top & B \end{smallmatrix}\right) \succeq 0, \;\; diag(A) \leq t, \quad diag(B) \leq t,$$

- we can rewrite as

$$\underset{A,B,X}{\text{minimize}} \left\{ \lambda \max diag \left(\begin{smallmatrix} A & X \\ X^\top & B \end{smallmatrix}\right) + \|P_\Omega(X) - P_\Omega(M)\|_F^2 + \delta_{\mathbf{S}_+^{m+n}} \left(\begin{smallmatrix} A & X \\ X^\top & B \end{smallmatrix}\right) \right\}$$

Applications: Matrix Completion



Objective value for PRISMA as a function of iterations in log scales
NCPP is a non-convex minimization approach for max-norm

Applications: Matrix Completion

	SDPT3		PRISMA		
Matrix Size	Cost	Time	Cost	Iter.	Time
100x100	1078.76	13s	1079.00	7366	124s
150x150	2617.65	89s	2618.11	7866	218s
200x200	4795.76	364s	4796.97	8908	281s
250x250	7736.76	1323s	7738.59	9524	404s
300x300	–	–	11406.39	9843	578s

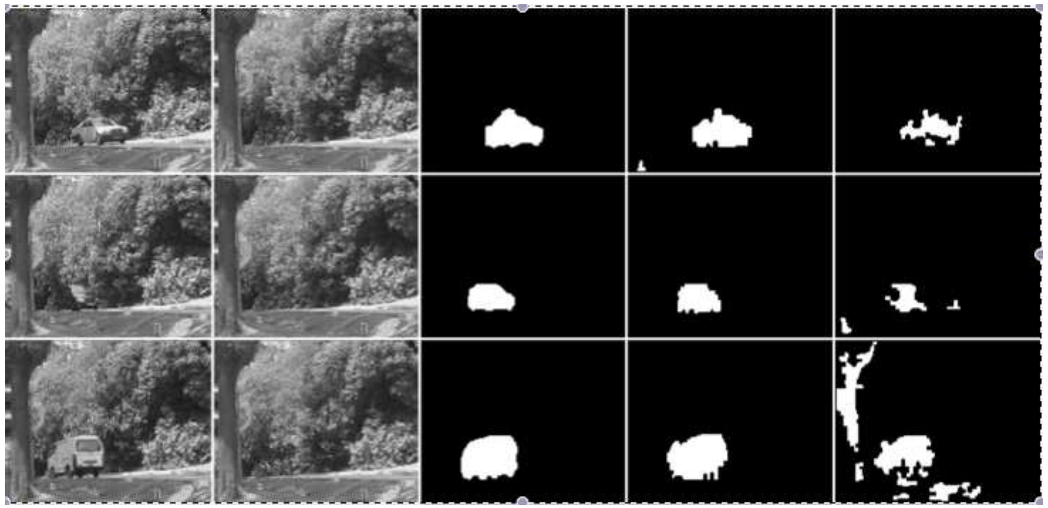
Applications: Robust PCA

- In the method of [Candes, Li, Ma & Wright] for robust PCA, a data matrix $M \in \mathbb{R}^{n_1 \times n_2}$ is given and the goal is to solve

$$\min\{\|W\|_{tr} + \lambda \|S\|_1 : W + S = M,\, W, S \in \mathbb{R}^{n_1 \times n_2}\}$$

- This problem is of the form covered by PRISMA with the choice $f = 0,\ g(W) = \lambda \|M - W\|_1,\ h(W) = \|W\|_{tr}$

Applications: Robust PCA

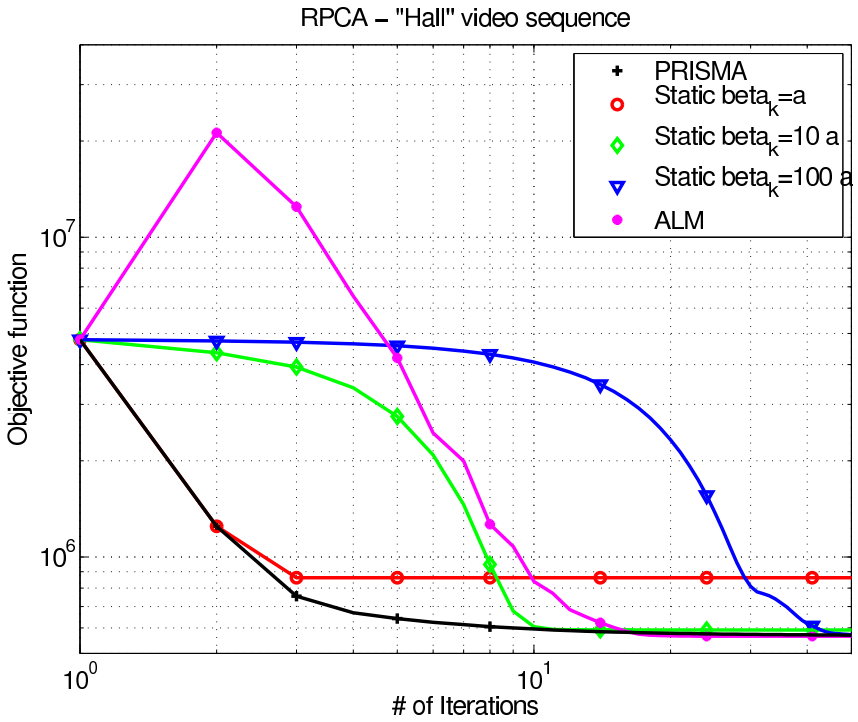


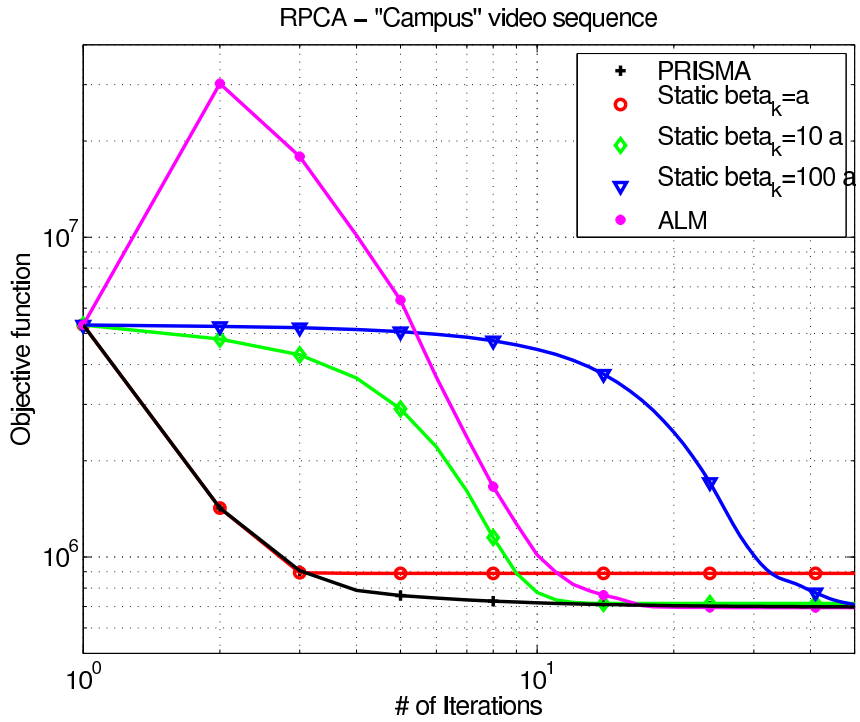
[Li,Huang,Gu & Tian]

	Campus	Hall
PRISMA (static & dynamic)	3.5 ± 0.1	1.9 ± 0.01
ALM	5 ± 0.3	2.66 ± 0.1

Time (in seconds) per step, average $\pm$ standard deviation

Applications: Robust PCA





Applications: Sparse Inverse Covariance Selection

- Goal is to solve

$$\min_{X \in \mathbf{S}^n_{++}} \; -\log\det(X) + \langle \Sigma, X \rangle + \lambda \|X\|_1$$

  where $\Sigma \in \mathbf{S}^n_+$ is a given positive semidefinite matrix

- This problem is of the form covered by PRISMA with the choice $f = \langle \Sigma, \cdot \rangle, g = \lambda \|\cdot\|_1$, $h = -\log\det(\cdot) + \delta_{\mathbf{S}^n_{++}}(\cdot)$

- The required prox_h can be computed with SVD

Applications: Sparse Inverse Covariance Selection

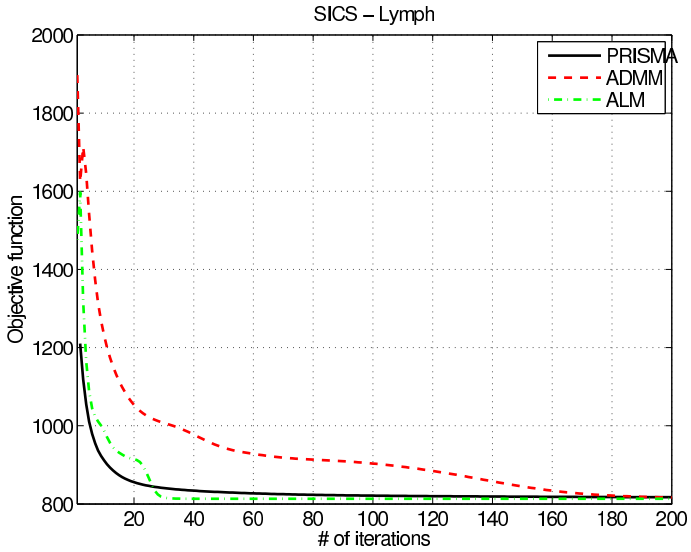
- Compared to ALM [Scheinberg et al. 2010], ADMM [Boyd et al. 2011]; ALM uses a continuation strategy with four parameters

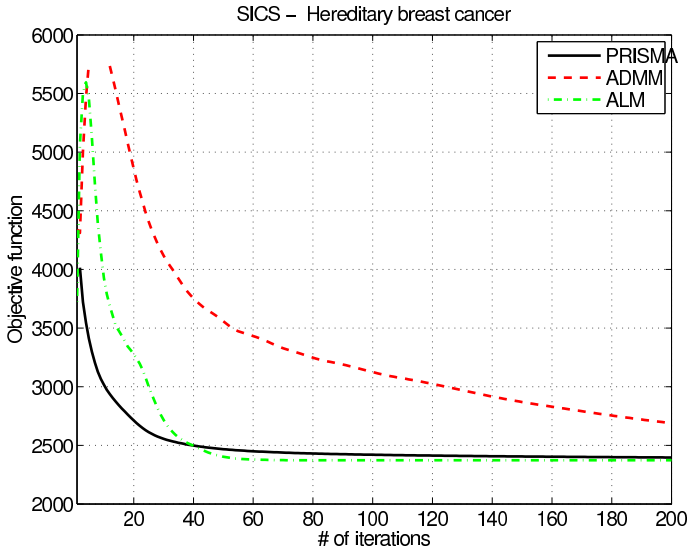
	Estrogen receptor	Arabidopsis thaliana	Leukemia
PRISMA	0.72 ± 0.04	1.15 ± 0.09	3.61 ± 0.19
ALM	0.45 ± 0.03	0.75 ± 0.04	2.46 ± 0.05
ADMM	0.80 ± 0.02	1.40 ± 0.05	4.15 ± 0.09

	Hereditary breast cancer	Lymph
PRISMA	11.74 ± 0.75	0.33 ± 0.02
ALM	7.5 ± 0.09	0.27 ± 0.05
ADMM	13 ± 0.23	0.47 ± 0.06

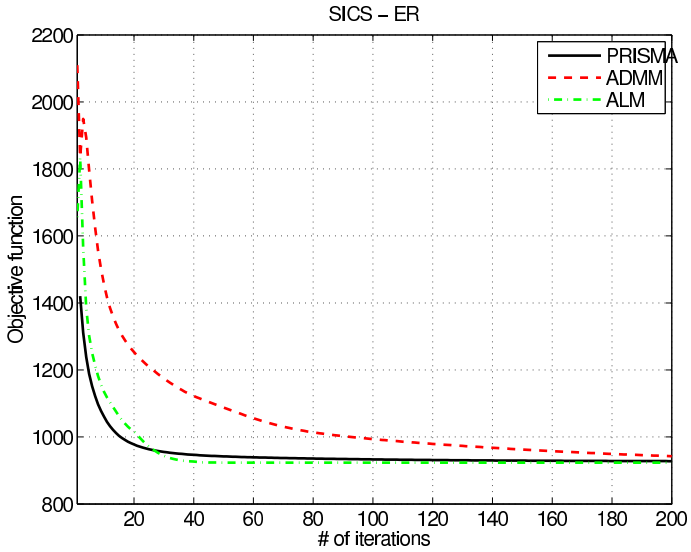
Time (in seconds) per step, average $\pm$ standard deviation

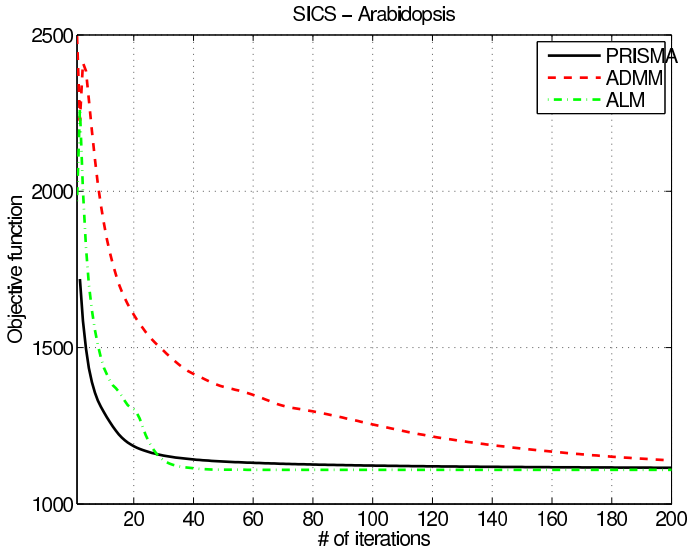
Applications: Sparse Inverse Covariance Selection



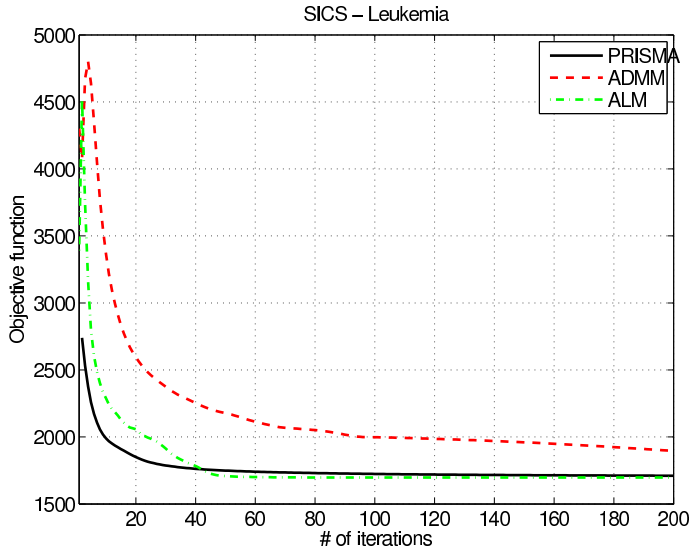


Applications: Sparse Inverse Covariance Selection





Applications: Sparse Inverse Covariance Selection



Remarks - Future Work

- Special case

$$\min\{g(\beta) + h(\beta) : \beta \in \mathbb{R}^d\}$$

  analogies to fwd-bwd, FISTA; but the rate is worse

- Goal: develop a general theory depending on analytical properties of g; what is the rate as a function of these properties?

- Can we prove *lower* bounds?