

Efficient Learning of Nonconvex Sparse and Low-rank Models

姚权铭

<http://www.cse.ust.hk/~qyaoaa/>

Covered Papers (mine)

- Large-Scale Low-Rank Matrix Learning with Nonconvex Regularizers. TPAMI. 2018.
- Efficient Learning with Nonconvex Regularizers by Nonconvexity Redistribution. JMLR. 2018.
- Efficient Learning with a Family of Nonconvex Regularizers by Redistributing Nonconvexity. ICML. 2016.
- Fast Low-Rank Matrix Learning with Nonconvex Regularization. ICDM. 2015.

Related Papers in Computer Vision (classical)

- Image denoising via sparse and redundant representations over learned dictionaries. TIP. 2006.
- Robust face recognition via sparse representation. TPAMI. 2009
- Robust subspace segmentation by low-rank representation. ICML. 2010.
- The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. Arxiv. 2010.
- Robust video denoising using low rank matrix completion. CVPR. 2010.
- Fast and accurate matrix completion via truncated nuclear norm regularization. TPAMI. 2013.
- A generalized iterated shrinkage algorithm for non-convex sparse coding. CVPR 2013.
- Weighted nuclear norm minimization with application to image denoising. ICCV. 2014.
- Generalized nonconvex nonsmooth low-rank minimization. CVPR. 2014.
- Hyperspectral image restoration using low-rank matrix recovery. TGRS. 2014

Related Papers in Computer Vision (recent)

- Scalable Robust Matrix Factorization with Nonconvex Loss. NIPS. 2018.
- Fast randomized singular value thresholding for nuclear norm minimization. TPAMI. 2018.
- Tensor Robust Principal Component Analysis with A New Tensor Nuclear Norm. TPAMI. 2018.
- Online Convolutional Sparse Coding with Sample-Dependent Dictionary. ICML. 2018.
- Nonlocal Low-Rank Tensor Factor Analysis for Image Restoration. CVPR. 2018.
- Efficient, Sparse Representation of Manifold Distance Matrices for Classical Scaling. CVPR. 2018.
- Efficient Low Rank Tensor Ring Completion. ICCV. 2017.
- Safe feature screening for generalized lasso. TPAMI. 2017.
- Non-convex Rank/Sparsity Regularization and Local Minima. ICCV. 2017.
- High Order Tensor Formulation for Convolutional Sparse Coding. ICCV. 2017.

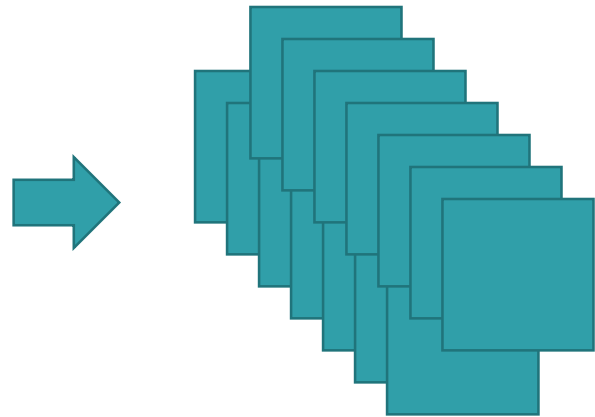
Agenda

- **Introduction**
 - **Example Applications**
 - **Nonconvex Regularization**
- **Preliminary : Proximal Gradient Algorithm**
- **N2C Transformation** (sparse)
- **FaNCL Algorithm** (low-rank)
- **Conclusion and Future Works**

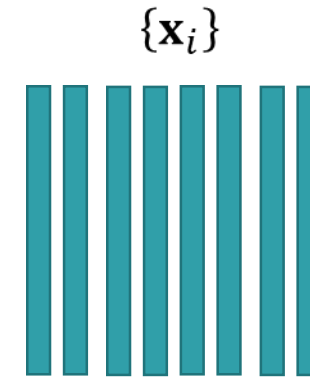
Sparse Coding for Image Denoising



Overlapping patches



Group similar patches and reshape into vectors



Sparse coding on these patches



Sparse Coding for Image Denoising

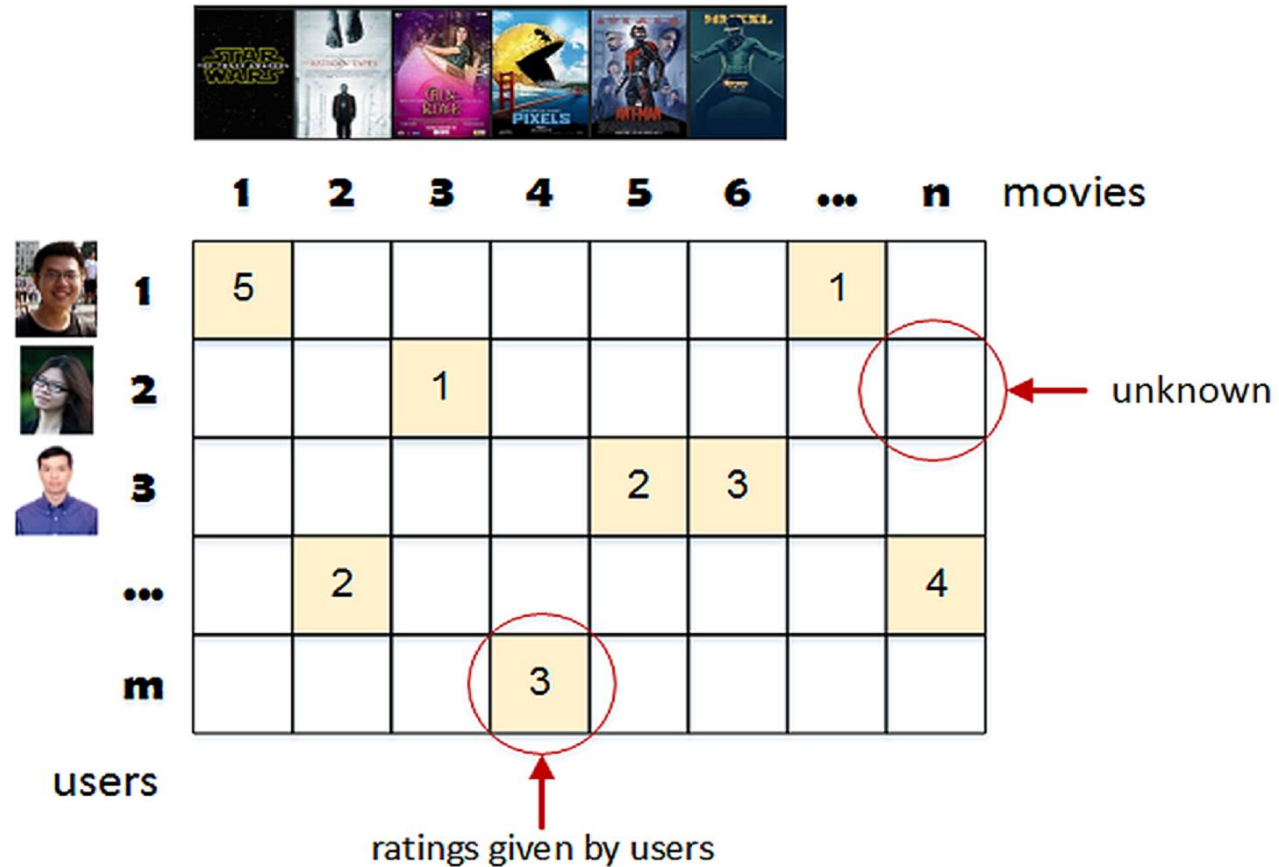
Objective:
$$\min_{\{\alpha_i\}} \frac{1}{2} \sum_{i=1}^N \underbrace{\| \mathbf{x}_i - \mathbf{D} \alpha_i \|_2^2}_{\text{loss}} + \lambda \underbrace{\| \alpha_i \|_1}_{\text{sparse regularization}}$$

- \mathbf{D} is dictionary learnt from noisy image/external data
- clean patch is recovered by $\mathbf{D} \alpha_i$

Highly redundant dictionary is key for good performance

- α_i is very sparse, convex l_1 -norm is popularly used to encouraging sparsity
- more complex structure can be encoded, e.g., group structure and tree-structure

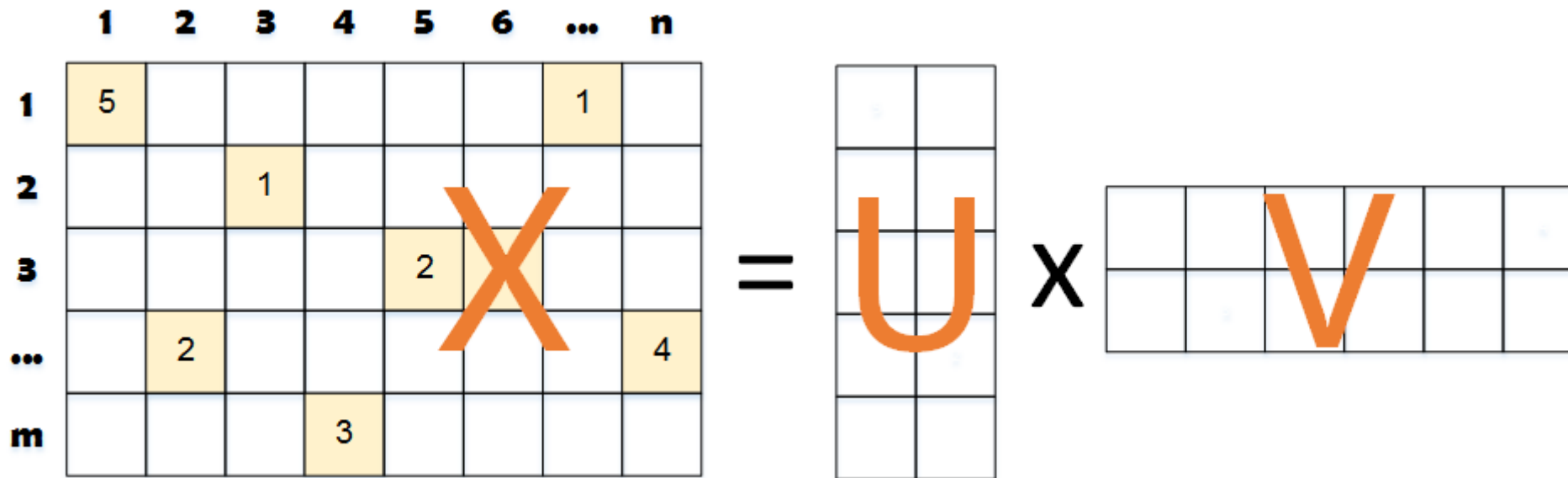
Matrix Completion for Recommender System



Predicting unknown ones based on existing observations

Matrix Completion for Recommender System

Similarity among users and items: low-rank assumption



Low-rank assumption $k \ll \min(m, n)$

- much less variables are needed \rightarrow capture relatedness and redundancy

Matrix Completion for Recommender System

Objective:
$$\min_{\{\mathbf{X}\}} \underbrace{\frac{1}{2} \sum_{i=1}^N \|P_{\Omega}(\mathbf{X} - \mathbf{O})\|_F^2}_{\text{loss}} + \underbrace{\lambda \|\mathbf{X}\|_*}_{\text{low-rank regularization}}$$

- P_{Ω} is a mask operator, if corresponding positions are not observed, their losses will be always set to zero
- $\|\mathbf{X}\|_* = \sum_{i=1}^m \sigma_i(\mathbf{X})$, where $\sigma_i(\mathbf{X})$ denotes i th singular value of \mathbf{X} (extension of l_1 -norm from vector to matrix case)

Capturing redundancy in \mathbf{O} is key for good performance

- if a matrix is of rank k , it has k nonzero singular values
- singular value threshold (SVD) is need to solve $\|\cdot\|_*$, which costs $O(m^2n)$ (very expensive)

Optimization Objectives

sparse: $\min_{\{\alpha_i\}} \frac{1}{2} \sum_{i=1}^N \|\mathbf{x}_i - \mathbf{D} \alpha_i\|_2^2 + \lambda \|\alpha_i\|_1$

low-rank: $\min_{\{\mathbf{X}\}} \frac{1}{2} \sum_{i=1}^N \|P_\Omega(\mathbf{X} - \mathbf{O})\|_F^2 + \lambda \|\mathbf{X}\|_*$



loss:
smooth

regularization:
nonsmooth

convex ones



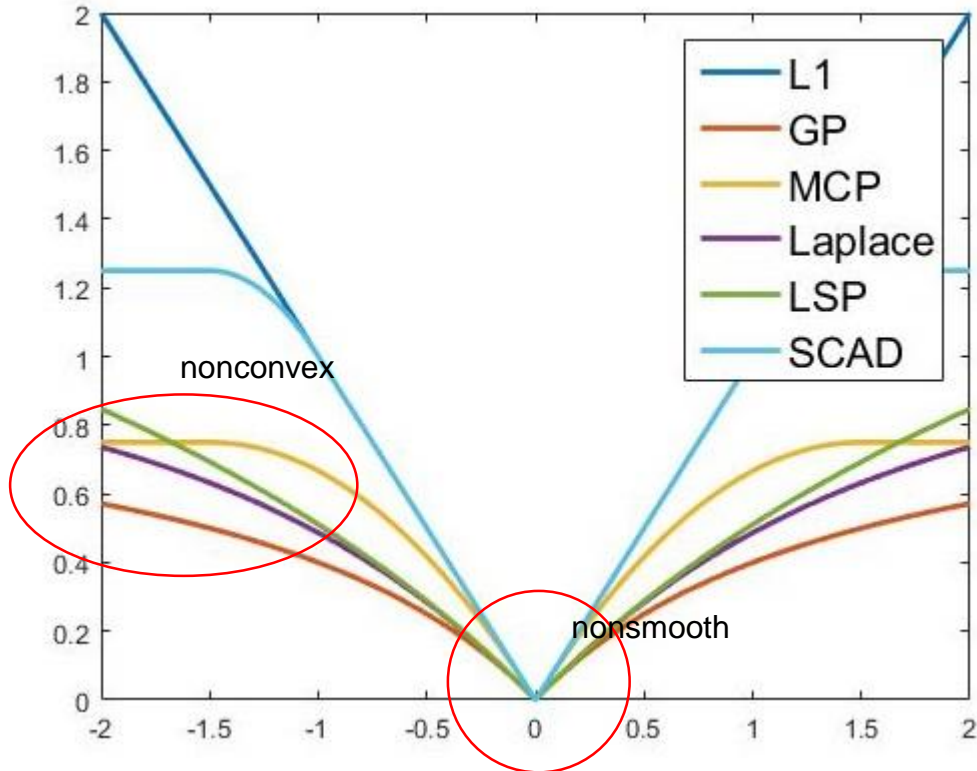
$$\min_{\mathbf{x}} f(\mathbf{x}) + \lambda g(\mathbf{x})$$

- f loss: smooth loss function
- g regularization: nonsmooth nonconvex

What's nonconvexity here?
Why we need nonconvex?

Nonconvex Regularization

One dimensional illustration



	$\kappa(\alpha)$	$\kappa'(\alpha)$	κ_0	ρ
GP (Geman and Yang, 1995)	$\frac{\beta\alpha}{\theta+\alpha}$	$\frac{\beta\theta}{(\theta+\alpha)^2}$	$\frac{\beta}{\theta}$	$\frac{2\beta}{\theta^2}$
LSP (Candès et al., 2008)	$\beta \log(1 + \frac{\alpha}{\theta})$	$\frac{\beta}{\theta+\alpha}$	$\frac{\beta}{\theta}$	$\frac{\beta}{\theta^2}$
MCP (Zhang, 2010a)	$\begin{cases} \beta\alpha - \frac{\alpha^2}{2\theta} & \alpha \leq \beta\theta \\ \frac{1}{2}\theta\beta^2 & \alpha > \beta\theta \end{cases}$	$\begin{cases} \beta - \frac{\alpha}{\theta} & \alpha \leq \beta\theta \\ 0 & \alpha > \beta\theta \end{cases}$	β	$\frac{1}{\theta}$
Laplace (Trzasko and Manduca, 2009)	$\beta(1 - \exp(-\frac{\alpha}{\theta}))$	$\frac{\beta}{\theta} \exp(-\frac{\alpha}{\theta})$	$\frac{\beta}{\theta}$	$\frac{\beta}{\theta^2}$
SCAD (Fan and Li, 2001)	$\begin{cases} \beta\alpha & \alpha \leq \beta \\ \frac{-\alpha^2+2\theta\beta\alpha-\beta^2}{2(\theta-1)} & \beta < \alpha \leq \theta\beta \\ \frac{\beta^2(1+\theta)}{2} & \alpha > \theta\beta \end{cases}$	$\begin{cases} \beta & \alpha \leq \beta \\ \frac{-\alpha+\theta\beta}{\theta-1} & \beta < \alpha \leq \theta\beta \\ 0 & \alpha > \theta\beta \end{cases}$	β	$\frac{1}{\theta-1}$

Table 1: Example nonconvex regularizers. Here, $\kappa_0 \equiv \kappa'(0)$ and $\beta > 0$. For SCAD, $\theta > 2$, whereas for others, $\theta > 0$.

- Features with large values are more informative, thus need to be less penalized
- All these penalties less penalize top features than the convex L1-norm (thus become nonconvex)

Nonconvex Regularization



(a)



(b)



(c)



(d)

much better performance

(a). original image, (b). blurry image, (c). deconvolution with L1, (d). with nonconvex regularization [from Zuo et al 2013]

regularizer	convex	nonconvex
optimization	😊 ✓	😞 ✗
performance	😞 ✗	😊 ✓

- convex reg leads to easy optimization but worse performance
- better performance can be obtained from nonconvex reg but optimization becomes much harder

Can we have best of both worlds?

Agenda

- Introduction
- **Preliminary : Proximal Gradient Algorithm**
 - Proximal step
- N2C Transformation (sparse)
- FaNCL Algorithm (low-rank)
- Conclusion and Future Works

Proximal Gradient (PG) Algorithm [Parikh & Boyd 2013]

PG algorithm is a general optimization framework solving

$$\min_{\mathbf{x}} f(\mathbf{x}) + \lambda g(\mathbf{x}) \quad \left\{ \begin{array}{l} \bullet f \text{ loss: smooth loss function} \\ \bullet g \text{ regularization: nonsmooth nonconvex} \end{array} \right.$$

In each iteration, it generates

$$\mathbf{x}_{t+1} = \arg \min_{\mathbf{x}} f(\mathbf{x}_t) + \langle \mathbf{x} - \mathbf{x}_t | \nabla f(\mathbf{x}_t) \rangle + \frac{L}{2} \|\mathbf{x} - \mathbf{x}_t\|_2^2 + \lambda g(\mathbf{x})$$

$$= \arg \min_{\mathbf{x}} \frac{1}{2} \left\| \mathbf{x} - \left(\mathbf{x}_t - \frac{1}{L} \nabla f(\mathbf{x}_t) \right) \right\|_2^2 + \lambda g(\mathbf{x}) \quad \longrightarrow$$

Key concept: Proximal Step

$$\mathbf{x}_{t+1} = \text{prox}_{\lambda g} \left(\mathbf{x} - \left(\mathbf{x}_t - \frac{1}{L} \nabla f(\mathbf{x}_t) \right) \right)$$

Proximal Gradient (PG) Algorithm [Parikh & Boyd 2013]

$$\mathbf{x} = \text{prox}_{\lambda g}(\mathbf{z}) \quad \longrightarrow \quad \text{cheap solutions}$$

- the sequence is iteratively generated from proximal step

When $g(\mathbf{x}) = \|\mathbf{x}\|_1$, the solution is called *soft-thresholding*, i.e., $[\text{prox}_{\lambda \|\cdot\|_1}(\mathbf{z})]_i = \text{sign}(\mathbf{x}_i) \cdot \max(|\mathbf{x}_i| - \lambda, 0)$

one pass of all features, cheap

When $g(\mathbf{x}) = \|\mathbf{X}\|_*$, the solution is called *singular value thresholding (SVT)*, i.e.,

$$\text{prox}_{\lambda \|\cdot\|_*}(\mathbf{Z}) = \mathbf{U}(\mathbf{\Sigma} - \lambda \mathbf{I})_+ \mathbf{V}^T$$

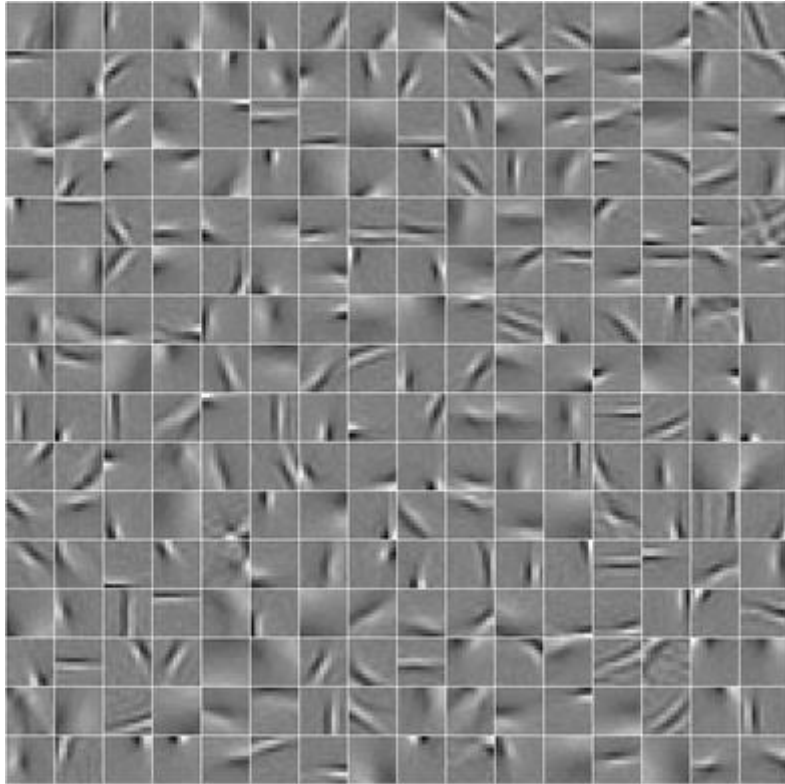
SVD is required, expensive

where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is SVD of \mathbf{Z} .

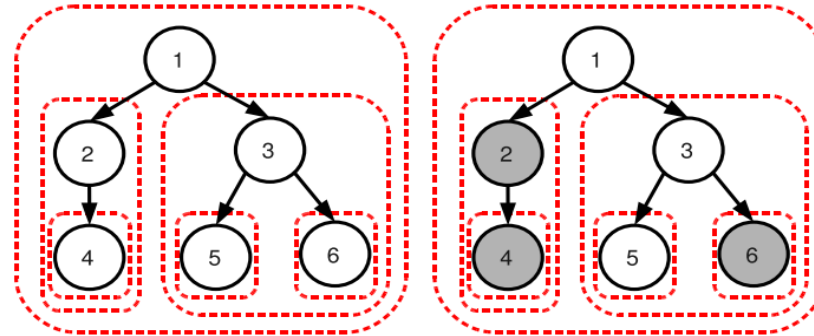
Agenda

- **Introduction**
- **Preliminary : Proximal Gradient Algorithm**
- **N2C Transformation** (sparse)
 - **Basic Idea**
 - **Use with PG**
 - **Experiments**
- **FaNCL Algorithm** (low-rank)
- **Conclusion and Future Works**

Tree Structured Lasso – An Example [Rodolphe et al 2011]

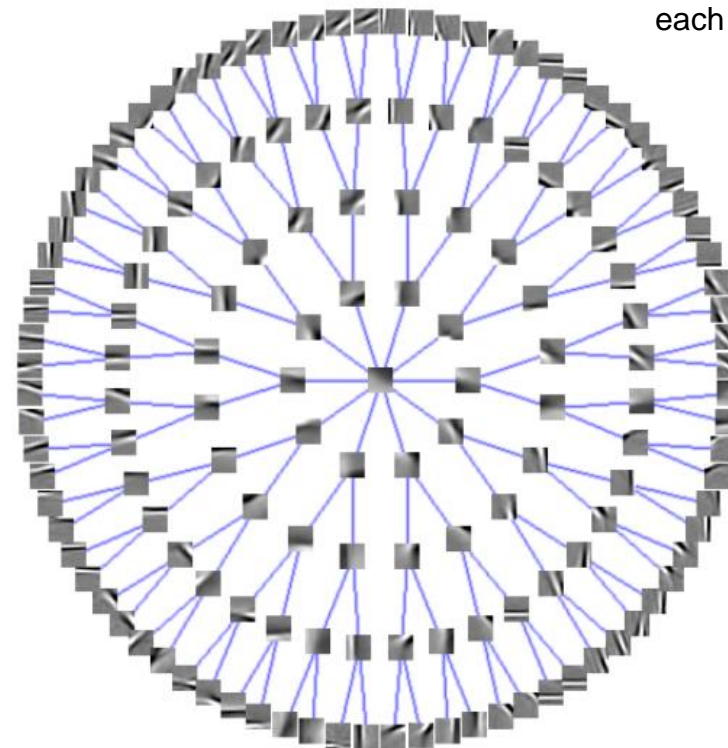


can be useful for analyzing patterns in data



How features can be organized

each red circle is one group



Example of the tree

Tree Structured Lasso – An Example [Rodolphe etal 2011]

$$\min_{\mathbf{x}} f(\mathbf{x}) + \lambda g(\mathbf{x})$$

$$\arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \lambda \|\mathbf{x}\|_1 + \sum_{j=1}^K \mu_j \|\mathbf{x}_{G_j}\|_2$$

one group

convex

cheaply solved by PG

$$\arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \lambda \sum_{i=1}^d \kappa(|x_i|) + \sum_{j=1}^K \mu_j \kappa(\|\mathbf{x}_{G_j}\|_2)$$

nonconvex

obtaining by using nonconvex function κ warp around convex norms

Proximal step is expensive

Transforming the objective!

- cheap closed-form for the convex case [Rodolphe etal 2011]
- **no closed-form**, iterative algorithms are needed for the **nonconvex one**

PG Algorithm – Look inside

$$\min_{\mathbf{x}} f(\mathbf{x}) + \lambda g(\mathbf{x})$$

To guarantee the convergence of PG, the most important thing is the smoothness of f .

Redistribute the nonconvexity from g to f

- still have the convergence guarantee
- has cheap proximal step as convex regularizers

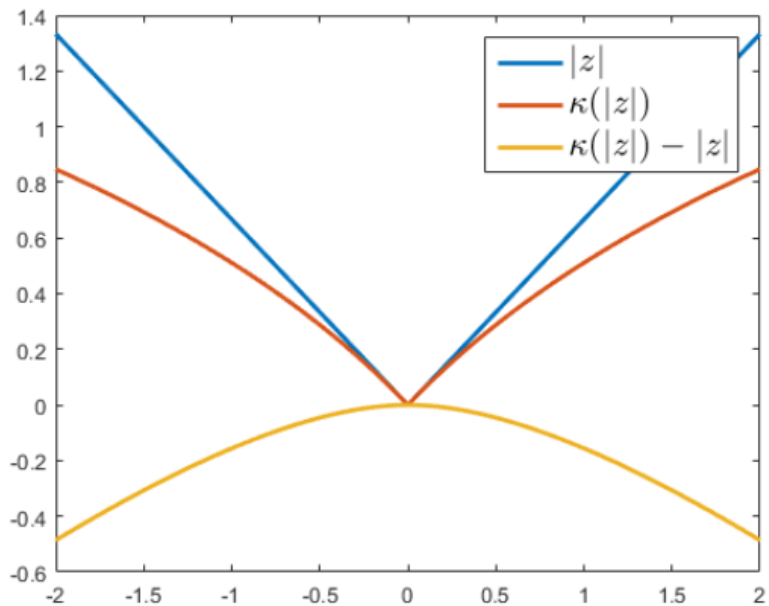
Lasso - Illustration

$$\text{Rewrite } g \text{ as } g(x) = \underbrace{\left(\sum_{i=1}^d \kappa(|x_i|) - \kappa_0 \|x\|_1 \right)}_{\bar{g}(x)} + \underbrace{\kappa_0 \|x\|_1}_{\check{g}(x)}$$

$z \in \mathbb{R}$, $\|z\|_p = |z|$, then

- $\kappa(|z|) - \kappa_0 |z|$ is smooth and concave

where $\kappa_0 = \kappa'(0)$



$\bar{g}(x)$

Smooth and Concave

$\check{g}(x)$

Convex

Nonconvex to Convex (N2C) Transformation

Problem $F(x) = f(x) + g(x)$ can then be rewritten as

$$F(x) = \bar{f}(x) + \check{g}(x),$$

where $\bar{f}(x) \equiv f(x) + \bar{g}(x)$

- \bar{f} (augmented loss): **smooth**, nonconvex
- \check{g} (convexified regularizer): **convex**, nonsmooth

Use PG on the Transformed Problem

Redistributing Nonconvexity:

- nonconvexity is shifted from regularizer to loss, while still ensuring that the augmented loss is smooth

Back to Tree Structured Lasso

After transformation (left)

$$\bar{f} + \check{g}$$

$$\arg \min_x \frac{1}{2} \|x - z\|_2^2 + \lambda \|x\|_1 + \sum_{j=1}^K \mu_j \|x_{G_j}\|_2$$

cheap closed-form

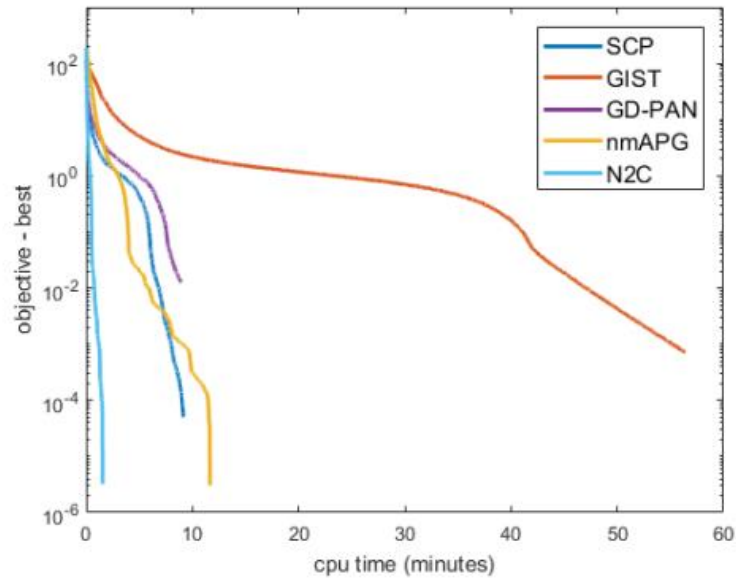
$$f + g$$

$$\arg \min_x \frac{1}{2} \|x - z\|_2^2 + \lambda \sum_{i=1}^d \kappa(|x_i|) + \sum_{j=1}^K \mu_j \kappa(\|x_{G_j}\|_2)$$

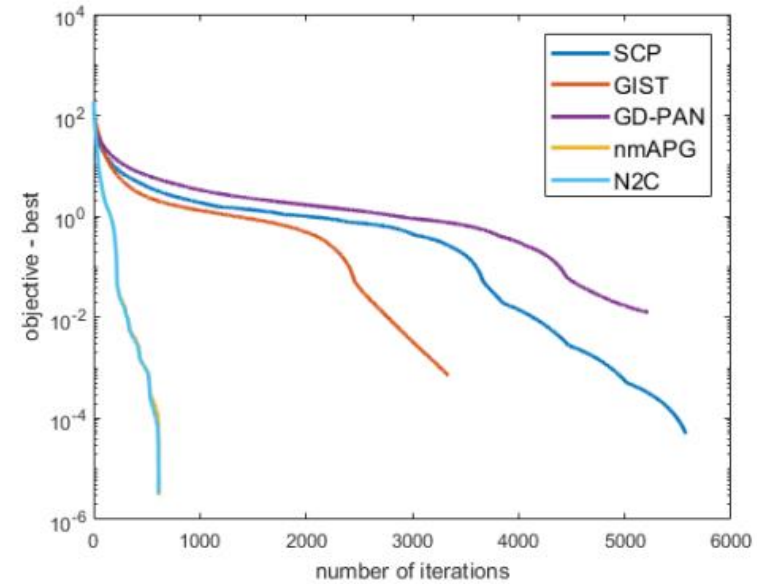
no closed-form, iterative algorithms are needed

Proximal algorithms on transformed problems can be very fast

Experiments



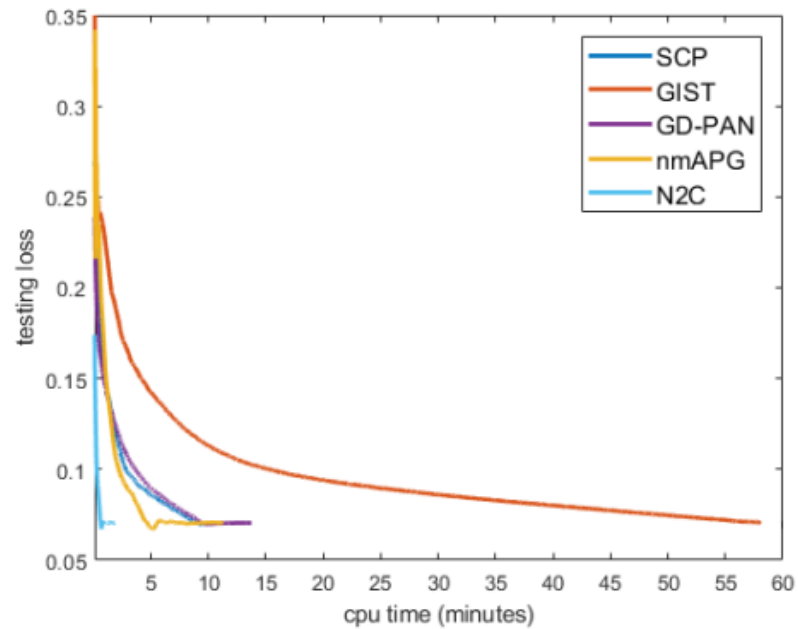
(a) Objective vs CPU time (minutes).



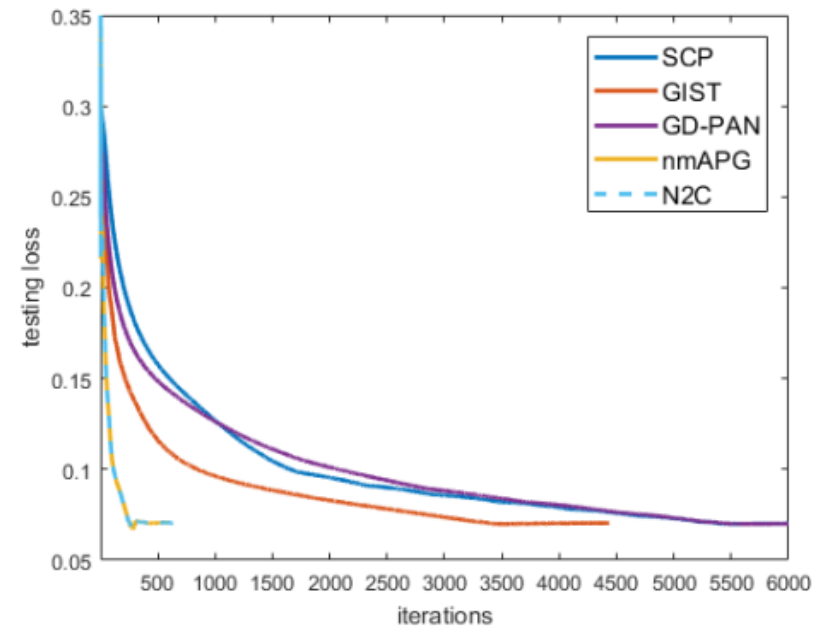
(b) Objective vs number of iterations.

Figure 8: Convergence of objective on nonconvex tree-structured group lasso. Note that the curves for nmAPG and N2C overlap in Figure 8(b).

Experiments



(a) Testing loss vs CPU time (minutes).



(b) Testing loss vs number of iterations.

Figure 9: Convergence of testing loss on nonconvex tree-structured group lasso.

Not Just PG

please check our JMLR paper

	section	advantages
proximal algorithm	4.1, 4.6	cheaper proximal step
FW algorithm	4.2	cheaper linear subproblem
(consensus) ADMM	4.3	cheaper proximal step; provide convergence guarantee
SVRG	4.4	cheaper proximal step; provide convergence guarantee
mOWL-QN	4.5	simpler analysis; capture curvature information

Table 2: Using the proposed convexification scheme with various algorithms.

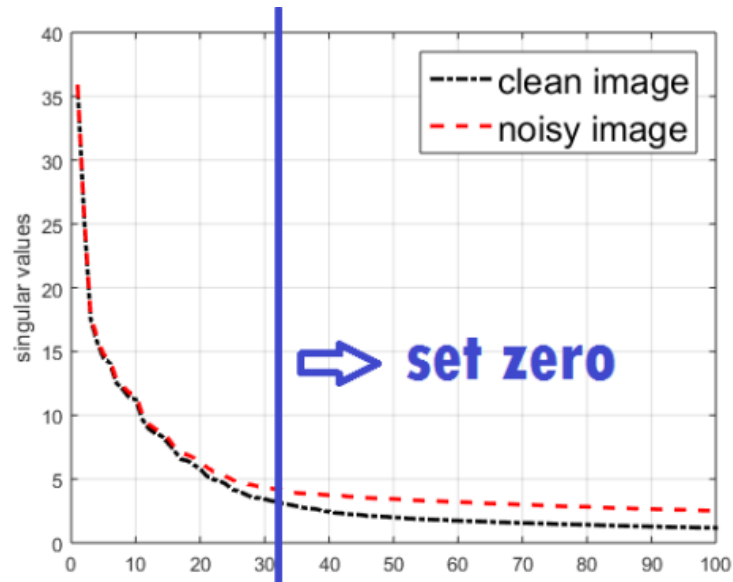
Whenever using a convex regularizer for sparse learning, you can

- use nonconvex penalty for better performance; and
- N2C to efficiently solve your new model

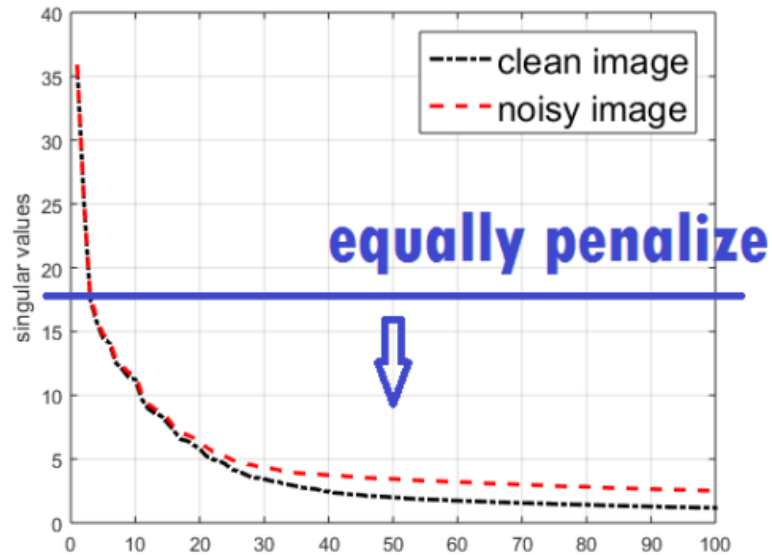
Agenda

- **Introduction**
- **Preliminary : Proximal Gradient Algorithm**
- **N2C Transformation** (sparse)
- **FaNCL Algorithm** (low-rank)
 - **Thresholding property**
 - **Sparse plus Low-rank Structure**
 - **Experiments**
- **Conclusion and Future Works**

Why we need Nonconvex Low-rank Regularizers?



(a) factorization



(b) nuclear norm

- factorization, $X = UV^T$:
set singular values outside selected rank to 0
- nuclear norm, $\|X\|_* = \sum_i \sigma_i(X)$:
equally penalize all singular values

What's the Problem for Matrix

$$\min_{\mathbf{x}} f(\mathbf{x}) + \lambda r(\mathbf{x})$$

nonconvex low-rank regularization

GSVT (Generalized Singular Value Thresholding operator) [Lu et al 2014]

The optimal solution of

$$\text{prox}_{\mu r}(Z) = \arg \min_X \frac{1}{2} \|X - Z\|_F + \lambda \sum_{i=1}^m \hat{r}(\sigma_i(X))$$

using nonconvex penalties on singular values

is $U \text{Diag}(y^*) V^T$, where $U \Sigma V^T$ is the SVD of Z , and $y^* = [y_i^*]$ with

$$y_i^* \in \arg \min_{y_i \geq 0} \frac{1}{2} (y_i - \sigma_i)^2 + \lambda \hat{r}(y_i)$$

PG algorithm can be directly used

- GSVT can be computed in closed-form using SVD $\rightarrow O(m^2n)$, **expensive**

Cut Down time on Proximal Step

FaNCL (Fast Nonconvex Low-rank algorithm)

How to cut down time complexity

- automatic thresholding property
- approximate SVD using power method
- further speedup with “sparse + low-rank” structure in matrix completion

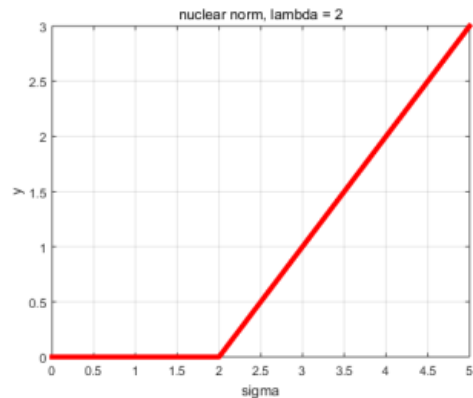
More than **100× faster** than state-of-art solvers and better performance than factorization & nuclear norm

Automatic Thresholding Property

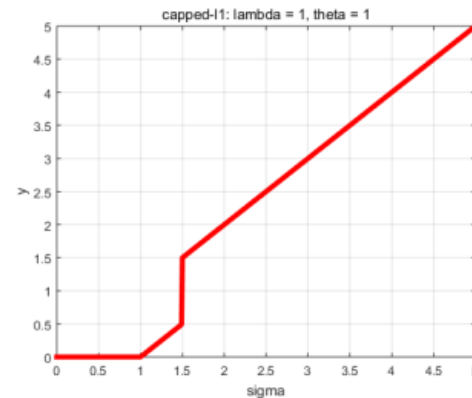
$$\text{prox}_{\lambda r}(Z) = U\text{Diag}(y^*)V^\top \longrightarrow y_i^* \in \arg \min_{y_i \geq 0} \frac{1}{2} (y_i - \sigma_i)^2 + \lambda \hat{r}(y_i)$$

Proposition (Automatic Thresholding)

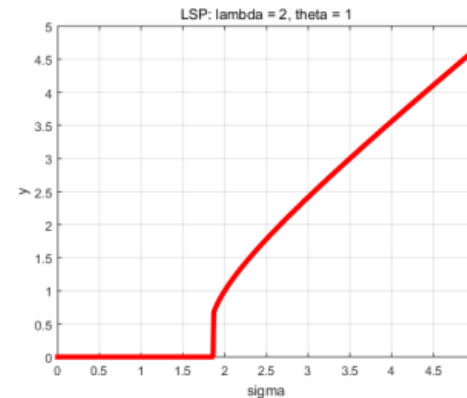
For any \hat{r} satisfying Assumption A3, there exists a threshold $\gamma > 0$ such that once $\sigma_i \leq \gamma$ then $y_i^* = 0$



(a) nuclear norm



(b) capped- ℓ_1



(c) LSP

Singular values are in non-ascending order, i.e. $\sigma_1 \geq \dots \geq \sigma_m$, once $\sigma_j \leq \gamma$ then for all $i \geq j$, $y_i^* = 0$

Allow partial SVD

Automatic Thresholding Property

Only top few singular values/vectors are needed \rightarrow approximate SVD by power method

Examples

- capped- ℓ_1 : $\gamma = \min(\mu, \theta + \frac{\mu}{2})$;
- LSP: $\gamma = \min(\frac{\mu}{\theta}, \theta)$;
- TNN: $\gamma = \max(\mu, \sigma_{\theta+1})$;
- SCAD: $\gamma = \mu$;
- MCP: $\gamma = \sqrt{\theta}\mu$ if $0 < \theta < 1$, and μ otherwise.

Approximate SVD using Power Method

Power Method [Halko et al., 2011]

Require: matrix $Z \in \mathbb{R}^{m \times n}$, $R \in \mathbb{R}^{n \times k}$.

```
1:  $Y^1 \leftarrow ZR$ ;  
2: for  $t = 1, 2, \dots, T_{pm}$  do  
3:    $Q^{t+1} = \text{QR}(Y^t)$ ; // QR decomposition  
4:    $Y^{t+1} = Z(Z^\top Q^{t+1})$ ;  
5: end for  
6:  $[U, \Sigma, V] = \text{SVD}((Q^{T_{pm}+1})^\top Z)$ ;  
7: return  $Q^{T_{pm}+1}U$ ,  $\Sigma$  and  $V$ .
```

- reduce from $O(m^2n)$ to $O(mnk)$
- further speedup to $O(\|\Omega\|_1 k)$ with “sparse + low rank” structure in matrix completion

$$Z^t = X^t - \frac{1}{\rho} \nabla f(X^t) = \underbrace{U^t V^{t\top}}_{\text{low-rank}} - \frac{1}{\rho} \underbrace{\mathcal{P}_\Omega(X^t - O)}_{\text{sparse}}$$

where X^t is maintained in factorized form, i.e. $X^t = U^t V^{t\top}$

Full Algorithm

FaNCL (Fast NonConvex Lowrank algorithm)

Require: $\tau > \rho$, $c_1 = \frac{\tau - \rho}{4}$, $\lambda^0 > \lambda$ and $\nu \in (0, 1)$;

- 1: randomly initialize $V_0, V_1 \in \mathbb{R}^{n \times k}$ and $X^1 = 0$;
- 2: **for** $t = 1, 2, \dots, T$ **do**
- 3: $\lambda^t \leftarrow (\lambda^{t-1} - \lambda)\nu + \lambda$;
- 4: $Z^t \leftarrow X^t - \frac{1}{\tau} \nabla f(X^t)$;
- 5: $V^{t-1} \leftarrow V^{t-1} - V^t(V^{t\top} V^{t-1})$, and remove any zero columns;
- 6: $R \leftarrow \text{QR}([V^t, V^{t-1}])$;
- 7: **for** $p = 1, 2, \dots$ **do**
- 8: $[\tilde{X}^p, R] = \text{ApproximateGSVT}(Z^t, R)$;
- 9: **if** $F(\tilde{X}^p) \leq F(X^t) - c_1 \|\tilde{X}^p - X^t\|_F^2$ **then**
- 10: $X^{t+1} \leftarrow \tilde{X}^p$, $V^{t+1} \leftarrow \tilde{V}^p$; **break**;
- 11: **else** $R^{p+1} = V_A^p$; **end if**
- 12: **end for**
- 13: **end for**
- 14: **return** X^{T+1} .

- step 8: approximate GSVD is done
- step 9: decreasing condition is checked, if it fails, improve approximation by repeatedly calling ApproximateGSVD

FaNCL - Convergence analysis

A limit point X^* can be obtained

Proposition

$$\sum_{t=1}^{\infty} \|X^{t+1} - X^t\|_F^2 < \infty.$$

The limit point is also a critical point

Theorem

$\{X^t\}$ converges to a critical point X^* of $F(X)$ in finite iterations.

Converge at $O(1/T)$ rate

Corollary

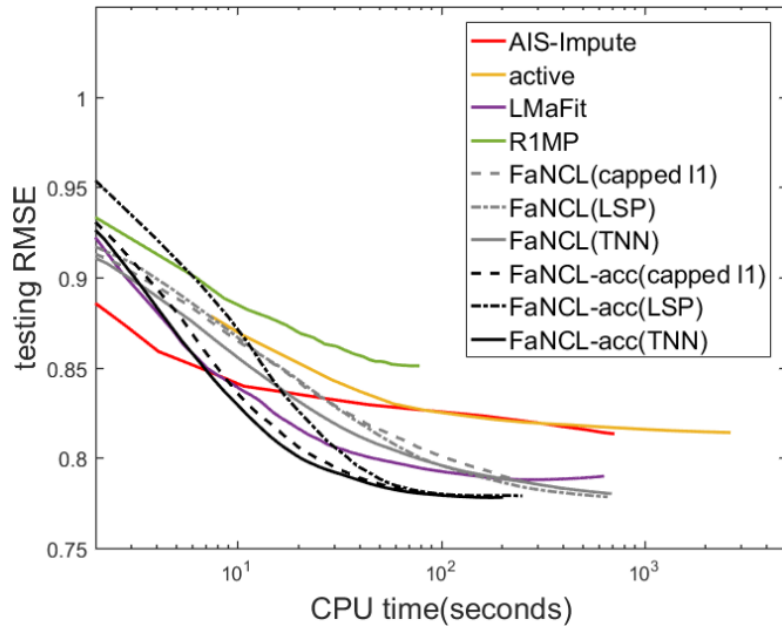
$$\min_{t=1, \dots, T} \|X^{t+1} - X^t\|_F^2 \leq \frac{1}{c_1 T} [F(X^1) - F(X^*)]$$

Can be extended to handle multiple blocks of parameters, such as RPCA

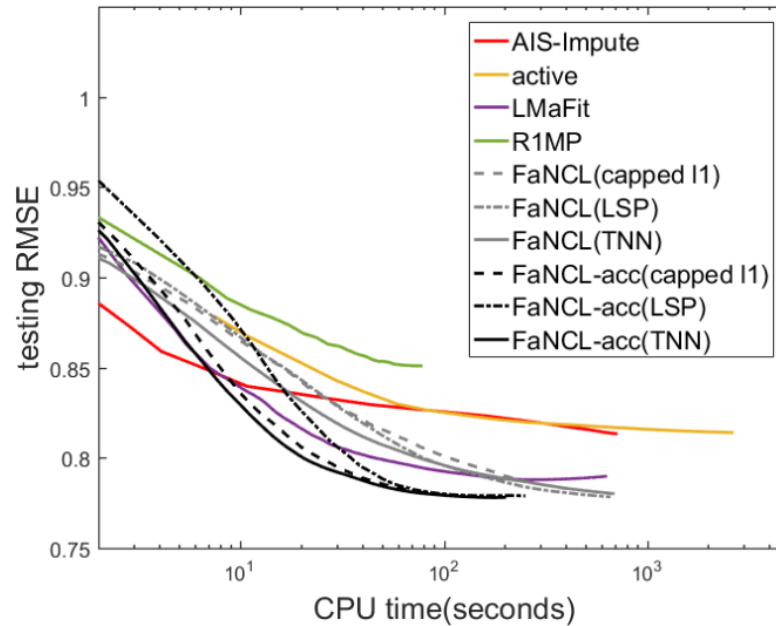
Experiments

		$m = 500$ (12.43%)			$m = 1000$ (6.91%)		
		NMSE	rank	time	NMSE	rank	time
nuclear norm	APG	4.26 ± 0.01	50	12.6 ± 0.7	4.27 ± 0.01	61	99.6 ± 9.1
	AIS-Impute	4.11 ± 0.01	55	5.8 ± 2.9	4.01 ± 0.03	57	37.9 ± 2.9
	active	5.37 ± 0.03	53	12.5 ± 1.0	6.63 ± 0.03	69	66.4 ± 3.3
fixed rank	LMaFit	3.08 ± 0.02	5	0.5 ± 0.1	3.02 ± 0.02	5	1.3 ± 0.1
	ER1MP	21.75 ± 0.05	40	0.3 ± 0.1	21.94 ± 0.09	54	0.8 ± 0.1
capped ℓ_1	IRNN	1.98 ± 0.01	5	14.5 ± 0.7	1.99 ± 0.01	5	146.0 ± 2.6
	GPG	1.98 ± 0.01	5	14.8 ± 0.9	1.99 ± 0.01	5	144.6 ± 3.1
	FaNCL	1.97 ± 0.01	5	0.3 ± 0.1	1.98 ± 0.01	5	1.0 ± 0.1
	FaNCL-acc	1.97 ± 0.01	5	0.1 ± 0.1	1.95 ± 0.01	5	0.5 ± 0.1
LSP	IRNN	1.96 ± 0.01	5	16.8 ± 0.6	1.89 ± 0.01	5	196.1 ± 3.9
	GPG	1.96 ± 0.01	5	16.5 ± 0.4	1.89 ± 0.01	5	193.4 ± 2.1
	FaNCL	1.96 ± 0.01	5	0.4 ± 0.1	1.89 ± 0.01	5	1.3 ± 0.1
	FaNCL-acc	1.96 ± 0.01	5	0.2 ± 0.1	1.89 ± 0.01	5	0.7 ± 0.1
TNN	IRNN	1.96 ± 0.01	5	18.8 ± 0.6	1.88 ± 0.01	5	223.1 ± 4.9
	GPG	1.96 ± 0.01	5	18.0 ± 0.6	1.88 ± 0.01	5	220.9 ± 4.5
	FaNCL	1.95 ± 0.01	5	0.4 ± 0.1	1.88 ± 0.01	5	1.4 ± 0.1
	FaNCL-acc	1.96 ± 0.01	5	0.2 ± 0.1	1.88 ± 0.01	5	0.8 ± 0.1

Experiments



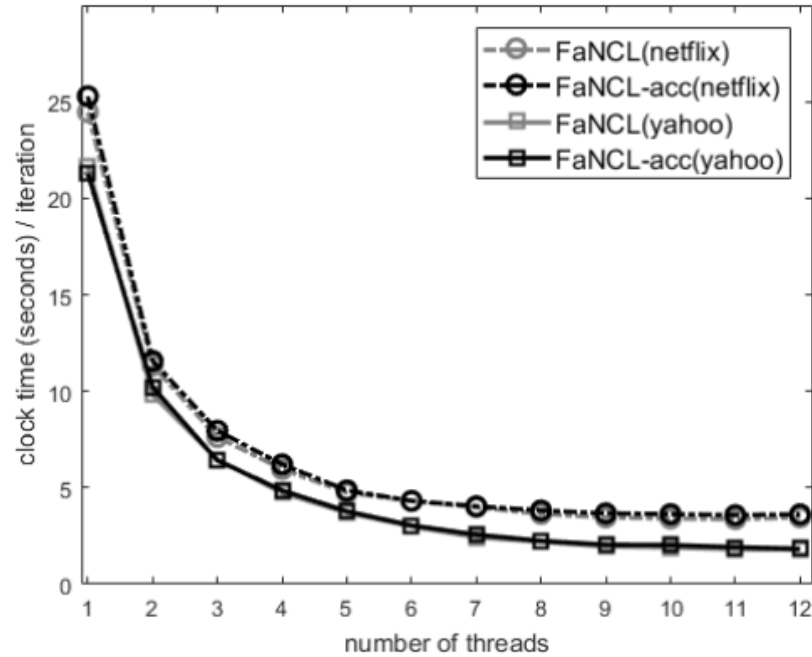
(a) 1M.



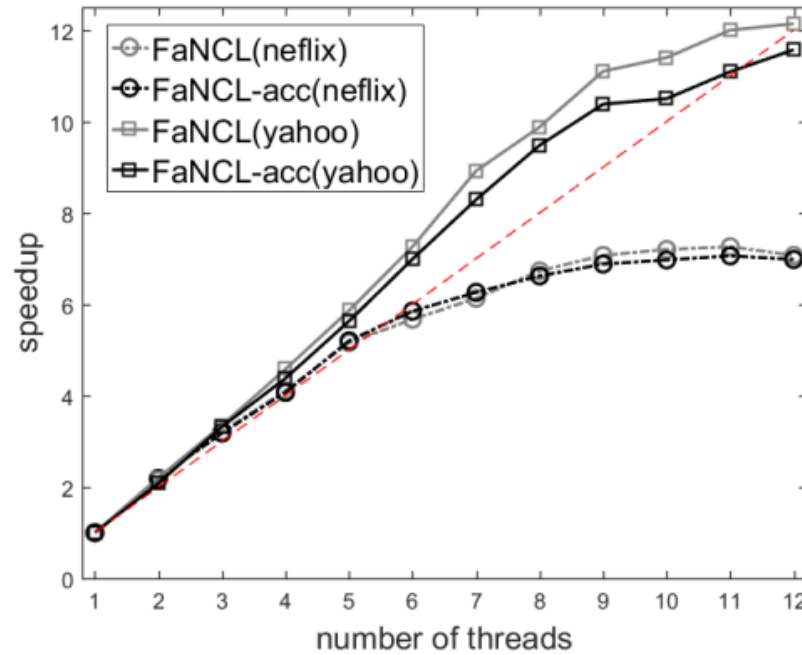
(b) 10M.

Fig. 5. RMSE vs CPU time on the *MovieLens-1M* and *10M* data sets.

Experiments – Parallel version



(a) Clock time per iteration.



(b) Speedup.

Summary

- Find the cut-off point and use partial SVD
- the SVD can be effectively approximated by power method
- Utilize problem structure

Agenda

- **Introduction**
- **Preliminary : Proximal Gradient Algorithm**
- **N2C Transformation** (sparse)
- **FaNCL Algorithm** (low-rank)
- **Conclusion and Future Works**

Conclusion

- Nonconvex penalties are useful to boost performance obtained from convex ones
- N2C is a powerful and general framework to learn sparse regularizers
- FaNCL is an efficient algorithm targeted at low-rank models
- Both N2C and FaNCL take the best from both worlds (efficiency from convex and performance from nonconvex)

Future Works

- Low-rank tensor learning
- Statistical performance of stationary points
- Automatic selection of proper nonconvex penalties