Scalable Structured Low Rank Matrix Optimization Problems

ROKS 2013

Marco Signoretto, ESAT-SCD/SISTA, KULeuven

joint work with V. Cevher and J. A. K. Suykens

Leuven July 10, 2013

Outline

Outline

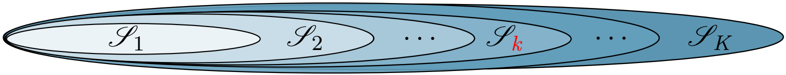
Learn From Empirical Data Via Regularization

Goal: find a model f from observational data

Learn From Empirical Data Via Regularization

Goal: find a model f from observational data

1. Construct nested subsets of increasingly complex hypotheses f



Learn From Empirical Data Via Regularization

Goal: find a model f from observational data

1. Construct nested subsets of increasingly complex hypotheses f

$$\mathscr{S}_k = \{f(x; w) \ : \ \Omega(w) \leq a_k\}$$

Learn From Empirical Data Via Regularization

Goal: find a model f from observational data

1. Construct nested subsets of increasingly complex hypotheses f

$$\mathscr{S}_k = \{f(x; w) \ : \ \Omega(w) \leq a_k\}$$

2. For each k, find an hypothesis that matches the data

Learn From Empirical Data Via Regularization

Goal: find a model f from observational data

1. Construct nested subsets of increasingly complex hypotheses f

$$\mathscr{S}_k = \{f(x; w) \ : \ \Omega(w) \leq a_k\}$$

2. For each k, find an hypothesis that matches the data

$$\hat{w}^k = \arg\min R_{\mathsf{emp}}(w) + \lambda_k \Omega(w) \qquad (\lambda_k \leftrightarrow a_k)$$

Learn From Empirical Data Via Regularization

Goal: find a model f from observational data

1. Construct nested subsets of increasingly complex hypotheses f

$$\mathscr{S}_k = \{f(x; w) \ : \ \Omega(w) \leq a_k\}$$

2. For each k, find an hypothesis that matches the data

$$\hat{w}^k = \arg\min R_{\mathsf{emp}}(w) + \lambda_k \Omega(w) \qquad (\lambda_k \leftrightarrow a_k)$$

3. Pick the complexity/fidelity trade-off hypothesis $f(x; \hat{w}^k)$

Learn From Empirical Data Via Regularization

Goal: find a model f from observational data

1. Construct nested subsets of increasingly complex hypotheses f

$$\mathscr{S}_k = \{f(x; w) \ : \ \Omega(w) \leq a_k\}$$

2. For each k, find an hypothesis that matches the data

$$\hat{w}^k = \arg\min R_{\mathsf{emp}}(w) + \lambda_k \Omega(w) \qquad (\lambda_k \leftrightarrow a_k)$$

3. Pick the complexity/fidelity trade-off hypothesis $f(x; \hat{w}^k)$

design of $\mathscr{S}_1 \subset \mathscr{S}_2 \subset \cdots \subset \mathscr{S}_K$ $\Leftrightarrow$ **choice of penalty** Ω

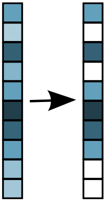
Structure-Inducing Penalties

prior knowledge: *sparsity*

l_1 penalty and the LASSO

$$\min_w R_{\text{emp}}(w) + \lambda \|w\|_1$$

- $w = [w_1; w_2; \cdots; w_P] \in \mathbb{R}^P$
- $f(x; w) = \langle x, w \rangle, \ \Omega(w) = \|w\|_1 = \sum_p |w_p|$

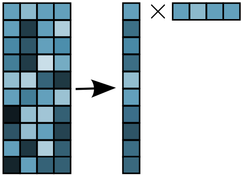


prior knowledge: *related tasks*

nuclear norm: multitask learning/collaborative filtering

$$\min_W \sum_t R_{\text{emp}}(w_t) + \lambda \|W\|_*$$

- $W = [w_1, \ldots, w_T] \in \mathbb{R}^{P \times T}, \ f_t(x; W) = \langle x, w_t \rangle$
- $\Omega(W) = \|W\|_* = \sum_r \sigma_r(W)$



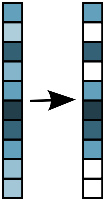
Composite Penalties

prior knowledge: *sparsity*

fused LASSO

$$\min_w R_{\text{emp}}(w) + \lambda \|Aw\|_1$$

- $w = [w_1; w_2; \cdots; w_P] \in \mathbb{R}^P$
- $\Omega(w) = \|Aw\|_1 = \sum_{p+1} |w_{p+1} - w_p|$

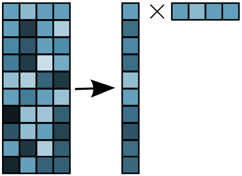


prior knowledge: *related tasks*

weighted nuclear norm

$$\min_W R_{\text{emp}}(W) + \lambda \|AWB^\top\|_*$$

- $W = [w_1, \ldots, w_T] \in \mathbb{R}^{P \times T}, \; f_t(x; W) = \langle x, w_t \rangle$



Outline

Structured Low-rank Learning Problem

Goal

Learn from observational data a matrix that, in addition to being low-rank, has entries partitioned into known disjointed groups.

Structured Low-rank Learning Problem

Goal

Learn from observational data a matrix that, in addition to being low-rank, has entries partitioned into known disjointed groups.

$$\min_{w \in \mathbb{R}^L} R_{\mathrm{emp}}(w) + \lambda \, \|\mathcal{B} w\|_*$$

- Composite (spectral) penalty
- Convex, can be turned into an SDP
- Structured matrix as the output of a *mutation* $\mathcal{B} : \mathbb{R}^L \to \mathbb{R}^{M \times N}$
- Nuclear norm used as a proxy for the rank

Encoding Group Structures via Mutations

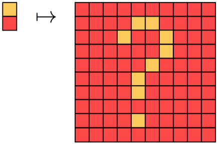
- Matrix entries partitioned into disjointed sets $\mathcal{P} = \{\mathscr{P}_1, \ldots, \mathscr{P}_L\}$

- Membership function associated to $\mathcal{P}$:

$$\begin{array}{rccc} \iota: & \mathbb{N}_M \times \mathbb{N}_N & \rightarrow & \mathbb{N}_L \\ & (m, n) & \mapsto & \{l \in \mathbb{N}_L \ : \ (m, n) \in \mathscr{P}_l\} \end{array}$$

- Mutation (forward) operator:

$$\begin{array}{rccc} \mathcal{B}: & \mathbb{R}^L & \rightarrow & \mathbb{R}^{M \times N} \\ & x & \mapsto & \left(x_{\iota(m,n)} \ : \ (m, n) \in \mathbb{N}_M \times \mathbb{N}_N\right) \end{array}$$



Application to System Identification

Goal: find a dynamical model from observed input and output signals

Nuclear Norm In Linear System Identification

- Motivated by well-known subspace properties
- Use of instrumental variables/matrix weights
- Modest improvement over classical subspace algorithms

Dealing with Missing Input and Output Observations

- Solve a structured low rank matrix optimization problem
- Reconstruct the system matrices via simple algebraic steps

Subspace Identification of Linear Dynamical Systems

State-space model of Order N_x

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

Realization Property

$$\mathcal{F} : (u, y) \mapsto \left[\frac{\mathcal{H}(u)}{\mathcal{H}(y)} \right], \ \mathcal{H}(x) = \begin{bmatrix} x(1) & x(2) & \cdots & x(T) \\ x(2) & x(3) & \cdots & x(T+1) \\ \vdots & \vdots & \ddots & \vdots \\ x(I) & x(I+1) & \cdots & x(T+I-1) \end{bmatrix}$$

$$\operatorname{rank}\left(\mathcal{F}(u, y)\right) = N_x + \operatorname{rank}(\mathcal{H}(u))$$

Subspace Identification of Linear Dynamical Systems

State-space model of Order N_x

$$\begin{cases} x(t+1) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) + Du(t) \end{cases}$$

System Identification with Missing Inputs and Outputs

$$\min_{u,y} \lambda_1 \left\| \mathcal{S}_u(u) - u_{\text{meas}} \right\|^2 + \lambda_2 \left\| \mathcal{S}_y(y) - y_{\text{meas}} \right\|^2 + \left\| \mathcal{F}(u,y) \right\|_*$$

[Z. Liu, A. Hansson, L. Vandenberghe, Nuclear norm system identification with missing inputs and outputs, *System and Control Letters* 62, 605-612, 2013]

Essentially a structured low rank matrix optimization problem

Outline

Proximal Algorithms for Nuclear-norm Problems

$$\min_{w} J(w) = R_{\text{emp}}(w) + \Omega(w)$$

Proximity Operator...

$$\text{prox}_{\Omega}(x) = \arg\min_{w} \Omega(w) + \frac{1}{2}\|x - w\|^2$$

Proximal Algorithms for Nuclear-norm Problems

$$\min_{w} J(w) = R_{\text{emp}}(w) + \Omega(w)$$

Forward-backward Splitting

$$w^{(k)} = \text{prox}_{\gamma\Omega}\left(w^{(k-1)} - \gamma \nabla R_{\text{emp}}\left(w^{(k-1)}\right)\right), \ \gamma > 0$$

Proximal Algorithms for Nuclear-norm Problems

$$\min_w J(w) = R_{\text{emp}}(w) + \Omega(w)$$

Forward-backward Splitting

- ⊕ simple to implement
- ⊕ scalable
- ⊕ can be accelerated

Proximal Algorithms for Nuclear-norm Problems

$$\min_{w} J(w) = R_{\text{emp}}(w) + \Omega(w)$$

Forward-backward Splitting

- ⊕ simple to implement
- ⊕ scalable
- ⊖ CPU time depends on global iteration complexity

Proximal Algorithms for Nuclear-norm Problems

$$\min_{w} J(w) = R_{\text{emp}}(w) + \Omega(w)$$

Simple Nuclear Norm Penalty: $\Omega(\cdot) = \|\cdot\|_*$

$\text{prox}_{\gamma\Omega}(\cdot)$ is the *singular value soft-thresholding operator*:

$$\begin{aligned} \text{if} \quad & X = U \,\text{diag}(\{\sigma_r\}_{1\le r\le R})\, V^\top \\ \text{then} \quad & \text{prox}_{\gamma\Omega}(X) = U \,\text{diag}(\{\max(\sigma_r - \gamma, 0)\}_{1\le r\le R})\, V^\top \end{aligned}$$

Proximal Algorithms for Nuclear-norm Problems

$$\min_w J(w) = R_{\text{emp}}(w) + \Omega(w)$$

Simple Nuclear Norm Penalty: $\Omega(\cdot) = \|\cdot\|_*$

$\text{prox}_{\gamma\Omega}(\cdot)$ is the *singular value soft-thresholding operator*:

$$\begin{array}{lrcl} \text{if} & X & = & U \,\text{diag}(\{\sigma_r\}_{1\le r\le R})\, V^\top \\ \text{then} & \text{prox}_{\gamma\Omega}(X) & = & U \,\text{diag}(\{\max(\sigma_r - \gamma, 0)\}_{1\le r\le R})\, V^\top \end{array}$$

Composite Nuclear Norm Penalty: $\Omega(\cdot) = \|\mathcal{B}\cdot\|_*$

- in general, not proximable (needs to be solved iteratively)
- $J(w^{(k)}) - J^* = O(1/k^2)$ under conditions [M. Schmidt et al., 2011], [S. Villa et al., 2012]

Proximal Algorithms for Nuclear-norm Problems

$$\min_w J(w) = R_{\text{emp}}(w) + \Omega(w)$$

Simple Nuclear Norm Penalty: $\Omega(\cdot) = \|\cdot\|_*$

$\text{prox}_{\gamma\Omega}(\cdot)$ is the *singular value soft-thresholding operator*:

$$\begin{array}{lrcl} \text{if} & X & = & U\,\text{diag}(\{\sigma_r\}_{1\le r\le R})\,V^\top \\ \text{then} & \text{prox}_{\gamma\Omega}(X) & = & U\,\text{diag}(\{\max(\sigma_r - \gamma, 0)\}_{1\le r\le R})\,V^\top \end{array}$$

Composite Nuclear Norm Penalty: $\Omega(\cdot) = \|\mathcal{B}\cdot\|_*$

$$\text{prox}_{\gamma\Omega}(x) = \mathcal{B}^* \left(\text{prox}_{\gamma\|\cdot\|_*}(\mathcal{B}x)\right)$$

only valid for very special mutations $\mathcal{B}$!

Implementing Mutations via Linear Indexing

Forward Operator: $\mathcal{B} : x \mapsto \left(x_{\iota(m,n)} \ : \ (m,n) \in \mathbb{N}_M \times \mathbb{N}_N\right)$

```
function Y=forwardOp(x,linSets,sizeY)
   Y=zeros(sizeY);
   for i=1:numel(linSets)
       Y(linSets{i})=x(i);
   end
```

Backward Operator: $\mathcal{B}^* : C \mapsto \left(\sum_{(m,n)\in\mathcal{P}_l} c_{mn} \ : \ l \in \mathbb{N}_L\right)$

```
 function y=backwardOp(X,linSets)
   y=zeros(numel(linSets),1);
   for i=1:numel(linSets)
      y(i)=sum(X(linSetsi));
   end
```

Efficient implementations can be given for special structures (e.g. Hankel)

Constrained Problem Formulation

$$\min_{w} R_{\text{emp}}(w) + \lambda \|\mathcal{B}w\|_*$$

Equivalent Problem with Separable Objective Function

$$\begin{aligned} \min_{w,Y} \quad & R_{\text{emp}}(w) + \lambda \| Y \|_* \\ \text{subject to} \quad & \mathcal{B}w = Y \end{aligned}$$

Constrained Problem Formulation

$$\min_w R_{\text{emp}}(w) + \lambda \|\mathcal{B}w\|_*$$

Equivalent Problem with Separable Objective Function

$$\begin{aligned} \min_{w,Y} \quad & R_{\text{emp}}(w) + \lambda \|Y\|_* \\ \text{subject to} \quad & \mathcal{B}w = Y \end{aligned}$$

- can be solved by ADMM/Douglas Rachford splitting

[S. Boyd et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foundations and Trends in Machine Learning* 3(1), 1-122, 2011]

Constrained Problem Formulation

$$\min_{w} R_{\text{emp}}(w) + \lambda \|\mathcal{B}w\|_*$$

Equivalent Problem with Separable Objective Function

$$\begin{array}{rl} \min_{w,Y} & R_{\text{emp}}(w) + \lambda \|Y\|_* \\ \text{subject to} & \mathcal{B}w = Y \end{array}$$

- can be solved by ADMM/Douglas Rachford splitting
- singular value soft-thresholding operator at each iteration

Constrained Problem Formulation

$$\min_w R_{\text{emp}}(w) + \lambda \|\mathcal{B}w\|_*$$

Equivalent Problem with Separable Objective Function

$$\begin{array}{rl} \min_{w,Y} & R_{\text{emp}}(w) + \lambda \|Y\|_* \\ \text{subject to} & \mathcal{B}w = Y \end{array}$$

- can be solved by ADMM/Douglas Rachford splitting
- singular value soft-thresholding operator at each iteration
- adaptive tolerances and Augmenter Lagrangian parameter

[Z. Liu, A. Hansson, L. Vandenberghe, Nuclear norm system identification with missing inputs and outputs, *System and Control Letters* 62, 605-612, 2013]

SVD-free Solution Strategy

$$\min_w R_{\mathrm{emp}}(w) + \lambda \|\mathcal{B}w\|_*$$

Equivalent Problem with Separable Objective Function

$$\begin{array}{ll} \min_{w,U,V} & R_{\mathrm{emp}}(w) + \lambda/2 \left(\|U\|_F^2 + \|V\|_F^2\right) \\ \text{subject to} & \mathcal{B}w = UV^\top \end{array} \qquad (\star)$$

($\star$) using that: $\|Y\|_* = \min_{U,V\,:\,Y=UV^\top} \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right)$

[M. Signoretto , V. Cevher and J.A.K. Suykens, An SVD-free Approach to a Class of Structured Low Rank Matrix Optimization Problems with Application to System Identification, Int. Rep. 13-44, ESAT-SISTA, K.U.Leuven 2013]

SVD-free Solution Strategy

$$\min_{w} R_{\text{emp}}(w) + \lambda \|\mathcal{B}w\|_*$$

Equivalent Problem with Separable Objective Function

$$\begin{aligned} \min_{w, U, V} \quad & R_{\text{emp}}(w) + \lambda /2 \left(\| U\|_F^2 + \| V\|_F^2\right) \\ \text{subject to} \quad & \mathcal{B}w = UV^\top \end{aligned} \qquad (\star)$$

($\star$) using that: $\| Y\|_* = \min_{U,V \,:\, Y=UV^\top} \frac{1}{2} \left(\| U\|_F^2 + \| V\|_F^2\right)$

- non-convex smooth problem

SVD-free Solution Strategy

$$\min_{w} R_{\text{emp}}(w) + \lambda \|\mathcal{B}w\|_*$$

Equivalent Problem with Separable Objective Function

$$\begin{aligned} \min_{w,U,V} \quad & R_{\text{emp}}(w) + \lambda/2 \left(\|U\|_F^2 + \|V\|_F^2\right) \\ \text{subject to} \quad & \mathcal{B}w = UV^\top \end{aligned} \qquad (\star)$$

$(\star)$ using that: $\|Y\|_* = \min_{U,V\,:\,Y=UV^\top} \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right)$

- size of matrix factors *can* be constrained

SVD-free Solution Strategy

$$\min_{w} R_{\text{emp}}(w) + \lambda \|\mathcal{B}w\|_*$$

Equivalent Problem with Separable Objective Function

$$\begin{aligned} \min_{w,U,V} \quad & R_{\text{emp}}(w) + \lambda/2\left(\|U\|_F^2 + \|V\|_F^2\right) \\ \text{subject to} \quad & \mathcal{B}w = UV^\top \end{aligned} \qquad (\star)$$

$(\star)$ using that: $\|Y\|_* = \min_{U,V\,:\,Y=UV^\top} \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right)$

- optimality of the non-convex heuristic for problems related to $(\star)$

[B., Recht, M. Fazel and P. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, SIAM Rev., 2010]

Augmented Lagrangian Approach

Equivalent Problem with Separable Objective Function

$$\begin{array}{ll} \min_{w,U,V} & R_{\text{emp}}(w) + \lambda/2\,(\|U\|_F^2 + \|V\|_F^2) \\ \text{subject to} & \mathcal{B}w = UV^\top \end{array} \qquad (\star)$$

Main Iteration

$$\begin{cases} w^{(k+1)} := \arg\min_w \; L_\mu(w, U^{(k)}, V^{(k)}; Z^{(k)}) & (1) \\ U^{(k+1)} := \arg\min_U \; L_\mu(w^{(k+1)}, U^{(k)}, V^{(k)}; Z^{(k)}) & (2) \\ V^{(k+1)} := \arg\min_V \; L_\mu(w^{(k+1)}, U^{(k+1)}, V^{(k)}; Z^{(k)}) & (3) \\ Z^{(k+1)} := Z^{(k)} + \mu\left(\mathcal{B}(w^{(k+1)}) - U^{(k+1)}V^{(k+1)\top}\right) & (4) \end{cases}$$

- $L_\mu(\cdot)$ Lagrangian of $(\star)$
- $(1, 2, 3)$ systems of linear equations if $R_{\text{emp}}(w) = (w - x)^\top H(w - x)$

Experiments

System Identification with Missing Inputs and Outputs

$$\min_{u,y} \lambda_1 \left\|\mathcal{S}_u(u) - u_{\text{meas}}\right\|^2 + \lambda_2 \left\|\mathcal{S}_y(y) - y_{\text{meas}}\right\|^2 + \|F(u,y)\|_*$$

Experimental Setting

- random inputs: $u(t) \in \mathbb{R}^P, t = 1, 2, \ldots, T$
- randomly generated stable state-space models with order S
- $y(t) \in \mathbb{R}^M, t = 1, 2, \ldots, T$ corrupted by $\epsilon(t) \sim \mathcal{N}(0, \sigma^2)$
- $\lambda_1 = \lambda_2 = 1$

Experiments

Constrained Non-convex Formulation

$$\begin{array}{cc} \min_{w,U,V} & \frac{1}{2}(w-a)^\top H_\lambda (w-a) + \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right) \\ \text{subject to} & \mathcal{B}w = UV^\top \end{array}$$

Evaluation Metrics

obj val	:	attained value in the constrained formulation
feasibility	:	primal feasibility $\|Y - \mathcal{B}(x)\|_F / \|Y\|_F$
model fit	:	averaged identification performance
CPU time	:	time in seconds used by the process

Experiments

Results averaged over 20 MC runs; $V\% = 20,\ P = 2,\ O = 3,\ \sigma = 0.1$

	obj val	feasibility (10^{-3})	model fit	CPU time (s)	matrix size
	$M = 5,\ T = 1500$				
SVD-free**	1016.79	0.30	79	0.67	56 × 1493
SVD-based*	1017.36	0.87	79	5.61	
	$M = 15,\ T = 1500$				
SVD-free**	1166.05	0.51	75	1.96	136 × 1493
SVD-based*	1165.71	0.30	75	8.65	
	$M = 20,\ T = 4000$				
SVD-free**	2079.09	0.22	76	6.78	176 × 3993
SVD-based*	2081.47	0.86	76	47.01	
	$M = 40,\ T = 4000$				
SVD-free**	2501.61	0.39	70	22.32	336 × 3993
SVD-based*	2501.76	0.37	70	120.16	
	$M = 50,\ T = 10000$				
SVD-free**	4803.46	0.30	67	111.28	416 × 9993
SVD-based*	4803.18	0.92	67	825.77	

$\left[**\right.$ factors of unrestricted size in $\mathcal{B}w = UV^\top$ $\left.\right]$

Experiments

Results averaged over 20 MC runs; $V\% = 20,\ P = 2,\ O = 3,\ \sigma = 0.1$

	obj val	feasibility (10^{-3})	model fit	CPU time (s)	matrix size
	$M = 5,\ T = 1500$				
SVD-free**	1016.79	0.30	79	0.67	56 × 1493
SVD-based*	1017.36	0.87	79	5.61	
	$M = 15,\ T = 1500$				
SVD-free**	1166.05	0.51	75	1.96	136 × 1493
SVD-based*	1165.71	0.30	75	8.65	
	$M = 20,\ T = 4000$				
SVD-free**	2079.09	0.22	76	6.78	176 × 3993
SVD-based*	2081.47	0.86	76	47.01	
	$M = 40,\ T = 4000$				
SVD-free**	2501.61	0.39	70	22.32	336 × 3993
SVD-based*	2501.76	0.37	70	120.16	
	$M = 50,\ T = 10000$				
SVD-free**	4803.46	0.30	67	111.28	416 × 9993
SVD-based*	4803.18	0.92	67	825.77	

[* Z. Liu, A. Hansson, L. Vandenberghe, Nuclear norm system identification with missing inputs and outputs, *System and Control Letters* 62, 2013]

Experiments

Results averaged over 20 MC runs; $V\% = 20, \ P = 2, \ O = 3, \ \sigma = 0.1$

	obj val	feasibility (10^{-3})	model fit	CPU time (s)	matrix size
	$M = 5, \ T = 1500$				
SVD-free**	1016.79	0.30	79	0.67	56 × 1493
SVD-based*	1017.36	0.87	79	5.61	
	$M = 15, \ T = 1500$				
SVD-free**	1166.05	0.51	75	1.96	136 × 1493
SVD-based*	1165.71	0.30	75	8.65	
	$M = 20, \ T = 4000$				
SVD-free**	2079.09	0.22	76	6.78	176 × 3993
SVD-based*	2081.47	0.86	76	47.01	
	$M = 40, \ T = 4000$				
SVD-free**	2501.61	0.39	70	22.32	336 × 3993
SVD-based*	2501.76	0.37	70	120.16	
	$M = 50, \ T = 10000$				
SVD-free**	4803.46	0.30	67	111.28	416 × 9993
SVD-based*	4803.18	0.92	67	825.77	

[* `[U,S,V]=svd(X)` instead of `[U,S,V]=svd(X,'econ')` !]

Experiments (cont'd)

$\sigma = 0$

	obj val	feasibility (10^{-3})	model fit	CPU time (s)	matrix size
	$M = 80, \ T = 10000$				
SVD-free full	2991.40	0.07	94.71	400.69	656 × 9993
SVD-free ($\cdot \times 22$)	2991.44	0.07	94.74	140.50	
SVD-econ	2991.66	0.09	94.60	609.24	

$\sigma = 0.03$

	obj val	feasibility (10^{-3})	model fit	CPU time (s)	matrix size
	$M = 80, \ T = 10000$				
SVD-free full	3279.09	0.13	85.88	285.88	656 × 9993
SVD-free ($\cdot \times 22$)	3279.09	0.13	85.90	132.42	
SVD-econ	3279.08	0.13	85.82	670.67	

$\sigma = 0.1$

	obj val	feasibility (10^{-3})	model fit	CPU time (s)	matrix size
	$M = 80, \ T = 10000$				
SVD-free full	6114.21	0.87	64.63	220.18	656 × 9993
SVD-free ($\cdot \times 22$)	6177.62	0.13	60.93	200.7	
SVD-econ	6114.21	0.90	64.63	222.62	

Experiments (cont'd)

$\sigma = 0$

	obj val	feasibility (10^{-3})	model fit	CPU time (s)	matrix size
	$M = 80, \ T = 10000$				
SVD-free full	2991.40	0.07	94.71	400.69	656 × 9993
SVD-free ($\cdot \times 22$)	2991.44	0.07	94.74	140.50	
SVD-econ	2991.66	0.09	94.60	609.24	

$\sigma = 0.03$

	obj val	feasibility (10^{-3})	model fit	CPU time (s)	matrix size
	$M = 80, \ T = 10000$				
SVD-free full	3279.09	0.13	85.88	285.88	656 × 9993
SVD-free ($\cdot \times 22$)	3279.09	0.13	85.90	132.42	
SVD-econ	3279.08	0.13	85.82	670.67	

$\sigma = 0.2$

	obj val	feasibility (10^{-3})	model fit	CPU time (s)	matrix size
	$M = 80, \ T = 10000$				
SVD-free full	11718.84	0.17	61.94	295.79	656 × 9993
SVD-free ($\cdot \times 22$)	15718.70	4.3	37.72	1332.29	
SVD-econ	11718.65	0.65	61.93	409.43	

Conclusions

Summary

- Mutations & Structured Low-rank Learning Problem
- Application to System Identification with missing data
- Solution strategy based on explicit factors

New Directions/Open Problems

- Guaranteed solutions: the role of noise
- Further exploitation of the structure of mutations
- Other applications of mutation-induced structured matrices