

First order methods for manifolds with orthogonality constraints for constrained SVD problems

Laura Balzano

University of Michigan

IMA Optimization New Directions Workshop 2016

Low-rank Models and Manifold Optimization

Low-rank matrix models have several applications:

- communications and radar
- computer vision
- recommender systems
- environmental science.

We therefore often face optimization problems that require non-convex low-rank constraints.

$$\begin{aligned} & \underset{M}{\text{minimize}} && f(M) \\ & \text{subject to} && h(M) \leq \tau \\ & && M \text{ low-rank, or has orthogonal columns, etc} \end{aligned}$$

Low-rank Models and Manifold Optimization

$$\begin{aligned} & \underset{M}{\text{minimize}} && f(M) \\ & \text{subject to} && h(M) \leq \tau \\ & && M \text{ low-rank, or has orthogonal columns, etc} \end{aligned}$$

Low-rank constraints often form *smooth manifolds* in \mathbb{R}^n .

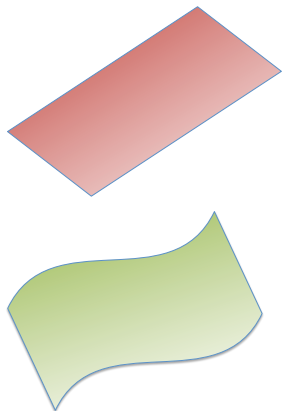
- Low-rank manifold: $n \times m$ matrices of rank- d
- Stiefel Manifold: tall $n \times d$ matrices with orthonormal columns denoted $\mathcal{S}(n, d)$.
- Grassmannian: d -dimensional subspaces of \mathbb{R}^n (quotient of Stiefel) denoted $\mathcal{G}(n, d)$.
- Flag Manifold: nested sequences of subspaces with given dimensions, $\mathcal{F}(n, d_1, \dots, d_s)$

Manifold Optimization

Classical optimization techniques use the geometry of Euclidean space.

Manifold optimization techniques use the geometry of the manifold to take gradients and gradient steps along the manifold.

We focus algorithms that use the Stiefel and Grassmann manifolds to solve “SVD-like” problems.



Outline

- The SVD as an optimization problem and variants where we may use manifold optimization
- Matrix Completion
- Robust PCA
- Sparse PCA
- Calibration Matrix Completion
- Conclusion

Subspace Model for Data

The Singular Value Decomposition (SVD) factors a low-rank matrix into three matrices:

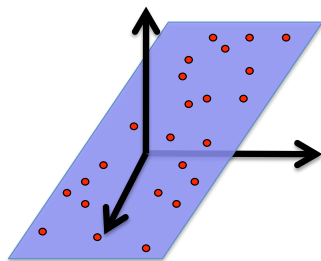
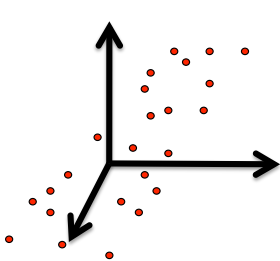
$$M = U\Sigma V^T$$

where U, V have orthonormal columns and Σ is diagonal.

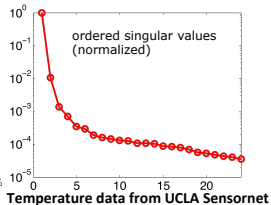
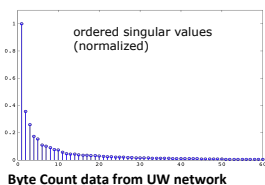
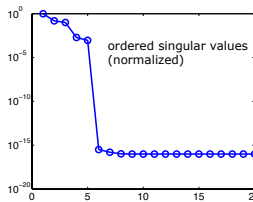
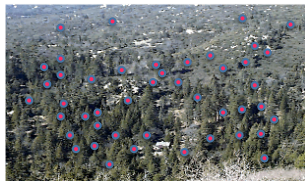
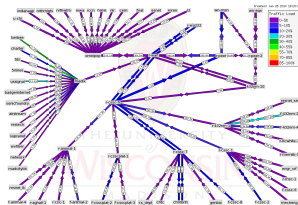


Linear Low Rank Subspaces via SVD

If your data matrix is not exactly low-rank but you wish to find the best low-dimensional linear structure that models the data, you can use the SVD; that makes it a useful exploratory data analysis tool.



The linear subspace is a good model in many applications.



Linear Low Rank Subspaces via SVD

The SVD gives the solution to the following problem¹:

$$\text{minimize } \|U\Sigma W^T - M\|_F^2 \quad (1)$$

$$\text{subject to } U, W \in \mathcal{G}(n, d) \\ \Sigma \succeq 0; d \times d \text{ diagonal} \quad (2)$$

where $\mathcal{G}(n, d)$ is the Grassmannian, the space of all d -dimensional subspaces of \mathbb{R}^n .

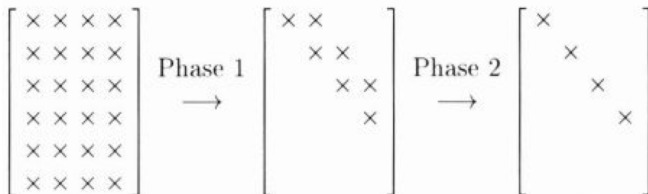
¹This result was discovered independently by both Schmidt in 1907 [Stewart, 1993, Stewart, 2011, Schmidt, 1907] and Eckart and Young in 1936 [Eckart and Young, 1936].

Linear Low Rank Subspaces via SVD

Goal: Given matrix A , form the SVD $U\Sigma V^T = X$.

The left singular vectors and singular values of A can be computed from the eigenvectors and the eigenvalues of AA^T . The right singular vectors from eigenvectors of $A^T A$. But the evals and evecs are hard to compute!

What we actually do is householder reflections to get a bidiagonal matrix, and versions of QR to get the SVD. $O(kn^3)$ operations for a square $n \times n$ matrix and k singular values/vectors.



The Incremental SVD for Euclidean Subspace Estimation

Goal: Given matrix $X = U\Sigma V^T$, form the SVD of $[X \quad v_t]$.

Estimate the weights: $w = \arg \min_a \|Ua - v_t\|_2^2$

Compute the residual: $r_t = v_t - Uw$.

Update the SVD:

$$[X \quad v_t] = \left[U \quad \frac{r_t}{\|r_t\|} \right] \begin{bmatrix} \Sigma & w \\ 0 & \|r_t\| \end{bmatrix} \begin{bmatrix} V & 0 \\ 0 & 1 \end{bmatrix}^T$$

and diagonalize the center matrix. What if we compute the same thing but with the partial-data weights and residual?

Adjustments to the SVD problem

If we add anything to the objective function, or add constraints, how do we adjust the SVD?

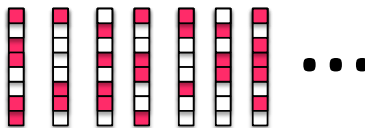
- An observation function (e.g., missing data) of the form $g(\cdot)$
- Any regularizer (e.g., ℓ_1 norm penalty) of the form $h(\cdot)$

$$\begin{aligned}
 & \underset{U \in \mathbb{R}^{n \times d}, W \in \mathbb{R}^{N \times d}}{\text{minimize}} && \|g(UW^T - M)\|_F^2 && (3) \\
 & \text{subject to} && h(g, U, W) \leq \tau \\
 & && U \in \mathcal{G}(n, d)
 \end{aligned}$$

With the lack of obvious extension to SVD computations, we turn to optimization!

Streaming SVD

Additionally we may observe our data streaming, again perhaps incomplete or under some other observation function.

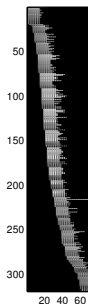
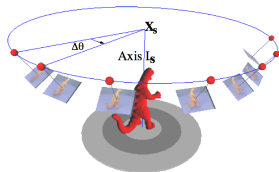
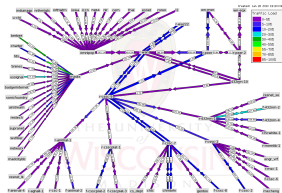


For these problems we focus on first order methods and relate them to ISVD.

Example Applications I

- ① Missing Data SVD (Low-Rank Matrix Completion):
 P_Ψ projects onto the coordinates $\Psi \subset \{1, \dots, n\}$.

$$\underset{U \in \mathbb{R}^{n \times d}, W \in \mathbb{R}^{N \times d}}{\text{minimize}} \quad \|P_\Psi(UW^T - M)\|_F^2 \quad \text{s.t. } U \in \mathcal{G}(n, d) \quad (4)$$



Example Applications II

② Robust SVD:

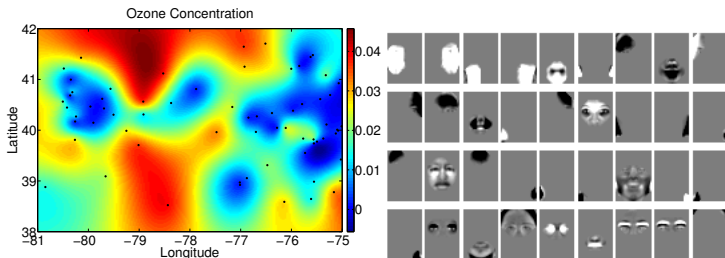
$$\begin{aligned} & \underset{U \in \mathbb{R}^{n \times d}, W \in \mathbb{R}^{N \times d}}{\text{minimize}} && \|P_{\Psi}(UW^T - M)\|_F^2 \\ & \text{s.t.} && \|UW^T - M\|_1 \leq \tau \text{ and } U \in \mathcal{G}(n, d) \quad (5) \end{aligned}$$



Example Applications III

3 Sparse SVD:

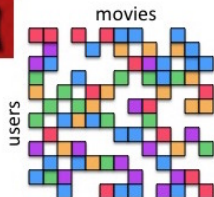
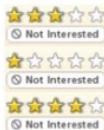
$$\underset{U \in \mathbb{R}^{n \times d}, W \in \mathbb{R}^{N \times d}}{\text{minimize}} \quad \|P_{\Psi}(UW^T - M)\|_F^2 \quad \text{s.t.} \quad \|U\|_1 \leq \tau, \quad U \in \mathcal{G} \quad (6)$$



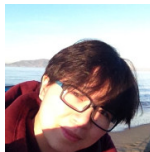
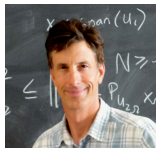
Example Applications IV

4 Calibration SVD:

$$\begin{aligned} & \underset{U \in \mathbb{R}^{n \times d}, W \in \mathbb{R}^{N \times d}, g}{\text{minimize}} && \|P_{\Psi} \left(g(UW^T + M) \right)\|_F^2 && (7) \\ & \text{subject to} && g \text{ is } L - \text{Lipschitz and monotonic} \\ & && U \in \mathcal{G}(n, d) \end{aligned}$$



Collaborators



Missing Data SVD (Low-Rank Matrix Completion)

$$\begin{aligned} & \underset{U \in \mathbb{R}^{n \times d}, W \in \mathbb{R}^{N \times d}}{\text{minimize}} && \|P_{\Psi}(UW^T - M)\|_F^2 \\ & \text{subject to} && U \in \mathcal{G}(n, d) \end{aligned}$$

P_{Ψ} projects onto the coordinates $\Psi \subset \{1, \dots, n\}$.

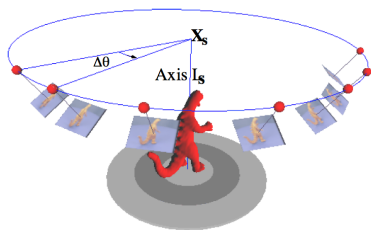
We could also replace P_{Ψ} with any compressed measurement matrix A .

Example 1: Structure from Motion

Observe an object from different camera angles, matching reference points on the object from image to image.

- Matrix of reference point locations for an orthographic camera in 2-d images has rank three, and the range subspace reveals 3-d location of reference points.
- Object is solid, so some reference points are occluded in each photo.

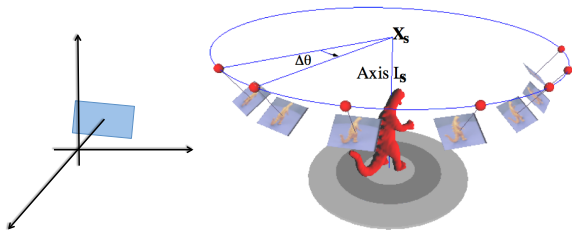
Missing data!



(Figure from Fitzgibbon, Cross, and Zisserman, 1998.)

Example 1: Structure from Motion

For example suppose one camera is aligned on the z-axis:



Then we could factorize:

$$\begin{bmatrix} x_{11} & y_{11} & \cdots & x_{1t} & y_{1t} \\ x_{21} & y_{21} & & x_{2t} & y_{2t} \\ \vdots & & & \vdots & \\ x_{n1} & y_{n1} & \cdots & x_{nt} & y_{nt} \end{bmatrix} = \begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix} \begin{bmatrix} 1 & 0 & c_{xx} & c_{yx} & \cdots \\ 0 & 1 & c_{yx} & c_{yy} & \cdots \\ 0 & 0 & c_{zx} & c_{zy} & \cdots \end{bmatrix}$$

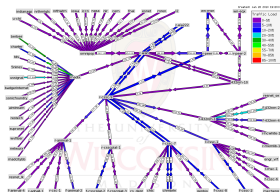
Structure from Motion: Figures and Reconstructions



(Kennedy, Balzano, Wright, Taylor [Kennedy et al., 2016])

Example 2: Computer network analysis


Byte counts are limited by the total number of source-destination flows in a computer network, and so they have a low-rank structure [Ding and Kolaczyk, 2013]. But the byte counts can not be streamed to a central location for every router, else they would congest the network.




Example 3: Recommender Systems

Netflix Prize Leaderboard

Mixture of
hundreds of
models, including
gradient descent



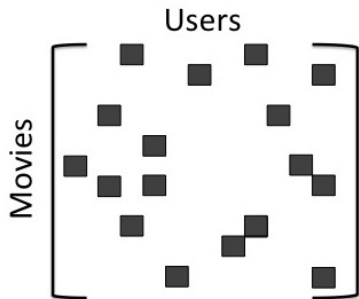
Gradient descent
on low-rank
parameterization



Rank	Team Name	Best Score	% Improvement	Last Submit Time
--	No Grand Prize candidates yet	--	--	--
Grand Prize - RMSE <= 0.8563				
--	No Progress Prize candidates yet	--	--	--
Progress Prize - RMSE <= 0.8625				
1	When Gravity and Dinosaurs Unite	0.8675	8.82	2008-03-01 07:03:35
2	ReReKor	0.8682	8.75	2008-02-28 23:40:45
3		0.8708	8.47	2008-02-06 14:12:44
Top 100 teams				
4	Lambert	0.8712	8.43	2007-10-01 23:25:23
5	acemah	0.8720	8.35	2008-03-02 05:08:12
6	Cris Tilters	0.8727	8.27	2008-03-02 00:42:29
7	hastor	0.8729	8.25	2007-11-24 14:27:00
8	Just a girl in a purple	0.8740	8.14	2008-02-06 12:16:40
9	BigBoys	0.8748	8.05	2008-03-01 17:28:08
10	Dinosaur Planet	0.8753	8.00	2007-10-04 04:58:45
..
50	wmgf	0.8897	6.48	2007-12-23 16:44:03
51	Remco	0.8899	6.46	2007-04-04 06:16:58
52	mfg	0.8900	6.45	2007-12-23 19:54:48
53	JustWithSVD	0.8900	6.45	2008-02-14 16:17:54
54		0.8900	6.45	2008-02-28 00:56:20
55		0.8901	6.44	2008-02-29 05:53:11
..
100	The_Crow	0.8902	6.43	2007-09-06 17:24:48

Low-rank Matrix Completion

We have an $n \times N$, rank r matrix X . However, we only observe a subset of the entries, $\Psi \subset \{1, \dots, n\} \times \{1, \dots, N\}$.



Low-rank Matrix Completion

We have an $n \times N$, rank r matrix X . However, we only observe a subset of the entries, $\Psi \subset \{1, \dots, n\} \times \{1, \dots, N\}$.

We may find a solution by solving the following NP-hard optimization:

$$\begin{aligned} & \underset{M}{\text{minimize}} \text{rank}(M) \\ & \text{subject to } M_{\Psi} = X_{\Psi} \end{aligned}$$

Low-rank Matrix Completion

We have an $n \times m$, rank r matrix X . However, we only observe a subset of the entries, $\Psi \subset \{1, \dots, n\} \times \{1, \dots, m\}$.

Or we may solve this convex problem:

$$\begin{aligned} \underset{M}{\text{minimize}} \quad & \|M\|_* = \sum_{i=1}^n \sigma_i(M) \\ \text{subject to} \quad & M_\Psi = X_\Psi \end{aligned}$$

Exact recovery guarantees: X is exactly low-rank and incoherent.
MSE guarantees: X is nearly low-rank with bounded $(r + 1)^{\text{th}}$ singular value.

Batch versus Online approach

We could solve the convex problem:

$$\underset{X \in \mathbb{R}^{n \times T}}{\text{minimize}} \quad \|X\|_* + \lambda \|P_\Psi(M - X)\|_F^2$$

Or we could solve this non-convex problem incrementally:

$$\underset{\substack{\text{span}(U) \in \mathcal{G}(d, n) \\ U^T U = I}}{\text{minimize}} \quad \|P_\Psi(UW^T - M)\|_F^2 = \sum_{t=1}^T \|P_{\Psi_t}(Uw - v_t)\|_2^2$$

Incremental Euclidean Optimization

$$\underset{U \in \mathbb{R}^{n \times d}}{\text{minimize}} \quad F(U) = \sum_{t=1}^T f_t(U)$$

Algorithm 1 Euclidean Incremental Gradient Descent

Given U_0 and step size regimen η_t

for $t = 1, 2, \dots, T$ **do**

 Compute the negative gradient at U_{t-1}

$$-\nabla f_t(U_{t-1}) = -\left. \frac{df_t}{dU} \right|_{U=U_{t-1}}$$

 Update: $U_t = U_{t-1} - \eta_t \nabla f_t(U_{t-1})$

end for



Incremental Grassmannian Optimization

$$\underset{\text{span}(U) \in \mathcal{G}(d,n)}{\text{minimize}} \quad F(U) = \sum_{t=1}^T f_t(U)$$

Algorithm 2 Grassmannian Incremental Gradient Descent

Given U_0 and step size regimen η_t

for $t = 1, 2, \dots, T$ **do**

 Compute SVD of negative gradient [Edelman et al., 1998]

$$-\nabla f_t(U_{t-1}) = -(I - UU^T) \frac{df_t}{dU} \Big|_{U=U_{t-1}} =: YSZ^T$$

 Update: $U_t = U_{t-1}Z \cos(S\eta_t)Z^T + Y \sin(S\eta_t)Z^T$

end for

Grassmannian Rank-One Update Subspace Estimation

$$\underset{\text{span}(U) \in \mathcal{G}(d,n)}{\text{minimize}} \quad F(U) = \sum_{t=1}^T \|P_{\Psi_t}(Uw - v_t)\|_2^2$$

Given current estimate U_t and partial data vector $P_{\Psi_t}(v_t)$, and a step size $\eta_t > 0$:

$$\begin{aligned} w_t &:= \arg \min_a \|P_{\Psi_t}(U_t a - v_t)\|_2^2; & p_t &:= U_t w_t; \\ P_{\Psi_t}(r_t) &:= P_{\Psi_t}(v_t - U_t w_t); & P_{\Psi_t^c}(r_t) &:= 0; \\ \sigma_t &:= \|r_t\| \|p_t\| \\ -\nabla f_t(U) &= \frac{r_t}{\|r_t\|} \sigma_t \frac{w_t^T}{\|w_t\|} \end{aligned}$$

Grassmannian Rank-One Update Subspace Estimation

$$\underset{\text{span}(U) \in \mathcal{G}(d,n)}{\text{minimize}} \quad F(U) = \sum_{t=1}^T \|P_{\Psi_t}(Uw - v_t)\|_2^2$$

Current estimate U_t , projection weights w_t , projection residual r_t , $p_t = U_t w_t$, $\sigma_t = \|r_t\| \|p_t\|$, and a step size $\eta_t > 0$:

$$-\nabla f_t(U) = \frac{r_t}{\|r_t\|} \sigma_t \frac{w_t^T}{\|w_t\|}$$

Then update:

$$U_{t+1} = U_t + \left[(\cos \sigma_t \eta_t - 1) \frac{p_t}{\|p_t\|} + \sin \sigma_t \eta_t \frac{r_t}{\|r_t\|} \right] \frac{w_t^T}{\|w_t\|}.$$

[Balzano et al., 2010]

GROUSE

- Process the v_t as a sequential stream.
- Cost function
$$f_t(U) = \min_a \|P_{\Psi_t}(U_t a - v_t)\|_2^2$$
- Perform incremental gradient descent constrained to the Grassmannian $\mathcal{G}(d, n)$.
- Maintain an $n \times d$ estimate U_t , with orthonormal columns, of the basis \bar{U} for target subspace \mathcal{S} .
- Simple update formula $U_t \rightarrow U_{t+1}$ when the next $[v_t]_{\Psi_t}$ is received.



Convergence definitions

Let $\phi_{t,i}$ represent the i^{th} principal angle between the true subspace \bar{U} and our estimate U_t . Then let:

$$\sum_{i=1}^d \sin^2(\phi_{t,i}) =: \epsilon_t$$

$$\prod_{i=1}^d \cos^2(\phi_{t,i}) =: \zeta_t$$

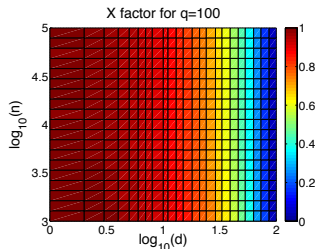
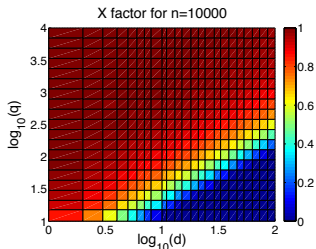
Suppose $v_t = \bar{U}s_t$ where s_t are identically distributed, zero-mean, and uncorrelated, and we have $|\Psi| = q$ measurements of each vector.

Calculating the Decrease Factor ('X') from simulations

Theorem ([Balzano and Wright, 2015])

In a local region around the global minimizer, we have a linear expected convergence rate:

$$\mathbb{E}[\epsilon_{t+1}|\epsilon_t] \leq \left(1 - X \frac{q}{nd}\right) \epsilon_t .$$



Convergence rate in noise

Theorem (GROUSE convergence in noise
[Zhang and Balzano, 2016])

Let the measurement be $v_t + n_t$ where $n_t \sim \mathcal{N}(0, \sigma^2)$.

$$\mathbb{E} [\zeta_{t+1} | \zeta_t] \geq \left(1 + \left(\frac{1}{1 + \frac{d}{n} \sigma^2} \right) \left(\frac{1 - \zeta_t}{d} \right) \left(1 - \frac{\sigma^2}{\frac{1 - \zeta_t}{d} + \sigma^2} \right) \right) \zeta_t$$

Corollary (No noise)

$$\mathbb{E} [\zeta_{t+1} | \zeta_t] \geq \left(1 + \frac{1 - \zeta_t}{d} \right) \zeta_t$$

Convergence

Theorem (GROUSE convergence from compressed data [Zhang and Balzano, 2016])

Let the measurement be $A(v_t + n_t)$ where $n_t \sim \mathcal{N}(0, \sigma^2)$ and A is a $q \times n$ Gaussian random matrix.

$$\mathbb{E}[\zeta_{t+1}|\zeta_t] \geq \left(1 + \beta_1 \frac{q}{n} \left(\frac{1 - \zeta_t}{d}\right) \left(1 - \frac{\sigma^2}{\frac{1 - \zeta_t}{d} + \sigma^2}\right)\right) \zeta_t$$

Convergence

Theorem (Global Convergence of GROUSE with full data [Zhang and Balzano, 2016])

Let $\epsilon^* > 0$ be the desired accuracy of our estimated subspace. Then for any $\rho, \rho' > 0$ and with a random initialization, after

$$K \geq K_1 + K_2 = \left(\left(\frac{d^3}{\rho'} + 1 \right) \log \frac{(d - \rho')}{d} + \frac{d^2}{d - 1} \log \left(\frac{1}{\epsilon^* \rho} \right) \right)$$

iterations of GROUSE Algorithm,

$$\mathbb{P}(\epsilon_K \leq \epsilon^*) \geq 1 - \rho' - \rho. \quad (8)$$

The Incremental SVD for Euclidean Subspace Estimation

Given matrix $X = U\Sigma V^T$, form the SVD of $[X \ v_t]$.

Estimate the weights: $w = \arg \min_a \|Ua - v_t\|_2^2$

Compute the residual: $r_t = v_t - Uw$.

Update the SVD:

$$[X \ v_t] = \begin{bmatrix} U & \frac{r_t}{\|r_t\|} \end{bmatrix} \begin{bmatrix} \Sigma & w \\ 0 & \|r_t\| \end{bmatrix} \begin{bmatrix} V & 0 \\ 0 & 1 \end{bmatrix}^T$$

and diagonalize the center matrix. What if we compute the same thing but with the partial-data weights and residual?

The Incremental SVD with Missing Data

Given matrix $X = U\Sigma V^T$, form the SVD of $[X \quad v_t]$.

Estimate the weights: $w = \arg \min_a \|P_{\Psi_t}(U_t a - v_t)\|_2^2$.

Compute the residual: $r_t = v_t - U w$ on Ψ_t ; zero otherwise.

Update the SVD:

$$\left[\begin{array}{c|c} U & \frac{r_t}{\|r_t\|} \end{array} \right] \left[\begin{array}{c|c} \Sigma & w \\ \hline 0 & \|r_t\| \end{array} \right] \left[\begin{array}{c|c} V & 0 \\ \hline 0 & 1 \end{array} \right]^T$$

and diagonalize the center matrix.

Incremental SVD with Missing Data: SAGE GROUSE

Given matrix $X = U\Sigma V^T$, form the SVD of $[X \quad v_t]$.

Estimate the weights: $w = \arg \min_a \|P_{\Psi_t}(U_t a - v_t)\|_2^2$.

Compute the residual: $r_t = v_t - U w$ on Ψ_t ; zero otherwise.

Update the SVD:

$$\begin{bmatrix} U & \frac{r_t}{\|r_t\|} \end{bmatrix} \begin{bmatrix} \mathcal{I}_d & w \\ 0 & \|r_t\| \end{bmatrix} \begin{bmatrix} V & 0 \\ 0 & 1 \end{bmatrix}^T$$

and take the SVD of the center matrix. This is equivalent to the natural incremental gradient method on the Grassmannian (GROUSE) for a particular step size [Balzano and Wright, 2013].

isvd details

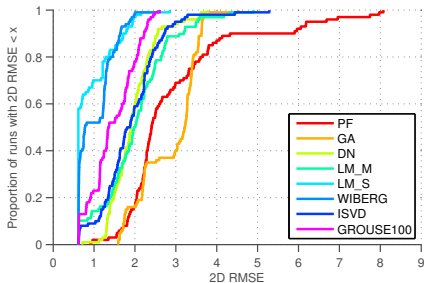


Structure from Motion: Figures and Reconstructions



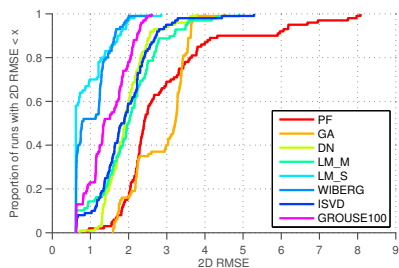
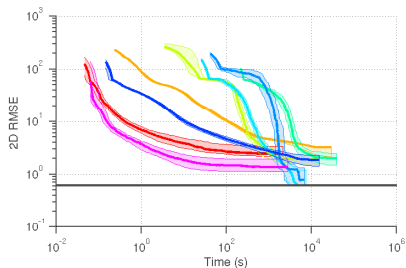
(Kennedy, Balzano, Wright, Taylor [Kennedy et al., 2016])

Structure from Motion: Figures and Reconstructions



(Kennedy, Balzano, Wright, Taylor [Kennedy et al., 2016])

Structure from Motion: Figures and Reconstructions



(Kennedy, Balzano, Wright, Taylor [Kennedy et al., 2016])

Robust PCA

$$\begin{aligned} & \underset{U \in \mathbb{R}^{n \times d}, W \in \mathbb{R}^{N \times d}}{\text{minimize}} && \|P_{\Psi}(UW^T - M)\|_F^2 \\ & \text{subject to} && \|UW^T - M\|_1 \leq \tau \\ & && U \in \mathcal{G}(n, d) \end{aligned}$$

Grassmannian Robust Adaptive Subspace Tracking Alg

$$\underset{\text{span}(U) \in \mathcal{G}(d,n)}{\text{minimize}} \quad F(U) = \sum_{t=1}^T \|s_t\|_1 + \|P_{\Psi_t}(Uw + s_t - v_t)\|_2^2$$

Current estimate U_t , projection weights w_t , proxy for projection residual considering sparse estimate Γ_t , $p_t = U_t w_t$, $\sigma_t = \|\Gamma_t\| \|p_t\|$, and a step size $\eta_t > 0$:

$$-\nabla f_t(U) = \frac{\Gamma_t}{\|\Gamma_t\|} \sigma_t \frac{w_t^T}{\|w_t\|}$$

Then update:

$$U_{t+1} = U_t + \left[(\cos \sigma_t \eta_t - 1) \frac{p_t}{\|p_t\|} + \sin \sigma_t \eta_t \frac{\Gamma_t}{\|\Gamma_t\|} \right] \frac{w_t^T}{\|w_t\|}.$$

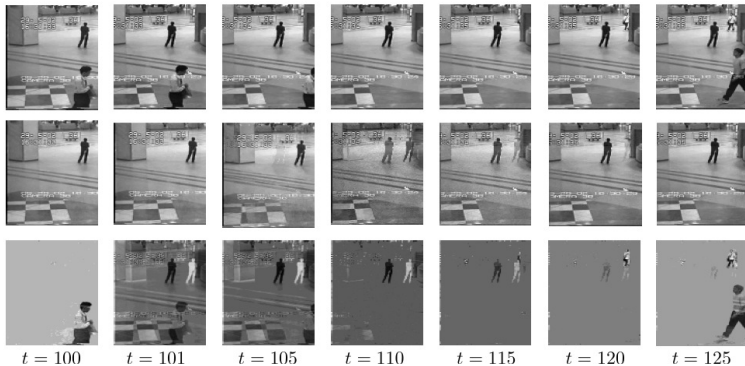
[He et al., 2012]

Foreground/background subtraction



Dataset	Resolution	Total Frames	Training Time	Tracking and Separating Time	FPS
Airport Hall	144×176	3584	11.3 sec	20.9 sec	171.5
Shopping Mall	320×256	1286	33.9 sec	27.5 sec	46.8
Lobby	144×176	1546	3.9 sec	71.3 sec	21.7
Hall with Virtual Pan (1)	144×88	3584	3.8 sec	191.3 sec	18.7
Hall with Virtual Pan (2)	144×88	3584	3.7 sec	144.8 sec	24.8

Foreground/background subtraction



Demo: Open CV code written by Arthur Szlam!

Sparse PCA

$$\begin{aligned} & \underset{U \in \mathbb{R}^{n \times d}, W \in \mathbb{R}^{N \times d}}{\text{minimize}} && \|P_{\Psi}(UW^T - M)\|_F^2 \\ & \text{subject to} && \|U\|_1 \leq \tau \\ & && U \in \mathcal{G}(n, d) \end{aligned}$$

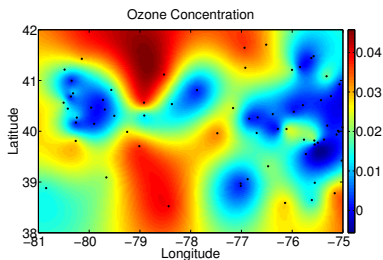
Example 1: Face image decomposition

Sparse PCA on Face data identifies salient parts of the image.



Example 2: Air pollution with spatial structure

Sparse PCA on ozone data identifies spatial regions that get influenced together.



SPCAur

$$\underset{\substack{\text{span}(U) \in \mathcal{G}(d,n) \\ R \in \mathcal{SO}(n)}}}{\text{minimize}} \quad F(U, R) = \lambda \|UR\|_1 + \sum_{t=1}^T \|P_{\Psi_t}(UW - v_t)\|_2^2$$

Current estimate U_t , and rotation R_t .

$$\frac{df_t}{dU} = \begin{cases} (U_{\Psi_t} w - v_{\Psi_t}) w^T + \lambda \cdot \text{sign}(U_t R_{t+1}) \cdot R_{t+1}^T & \text{on } \Psi_t \\ \lambda \cdot \text{sign}(U_t R_{t+1}) \cdot R_{t+1}^T & \text{on } \Psi_t^C \end{cases}$$

$$-\nabla f_t(U_{t-1}) = \left(I - U_t U_t^T \right) \frac{df_t}{dU} \Big|_{U=U_{t-1}} =: YSZ^T$$

SPCAur (continued)

$$-\nabla f_t(U_{t-1}) = \left(I - U_t U_t^T \right) \frac{df_t}{dU} \Big|_{U=U_{t-1}} =: YSZ^T$$

Then update [Edelman et al., 1998]:

$$U_{t+1} = [U_t Z \quad Y] \begin{bmatrix} \cos(\eta_t S) \\ \sin(\eta_t S) \end{bmatrix} Z^T \quad (9)$$

Fix U_{t+1} and update R_{t+1} on the Stiefel manifold.

Sparse PCA Results

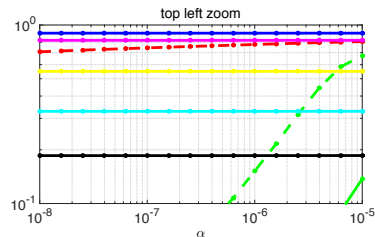
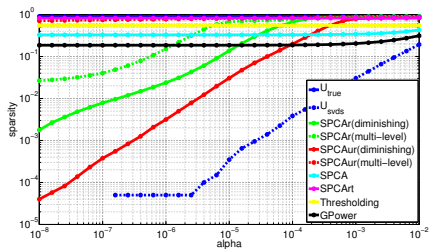


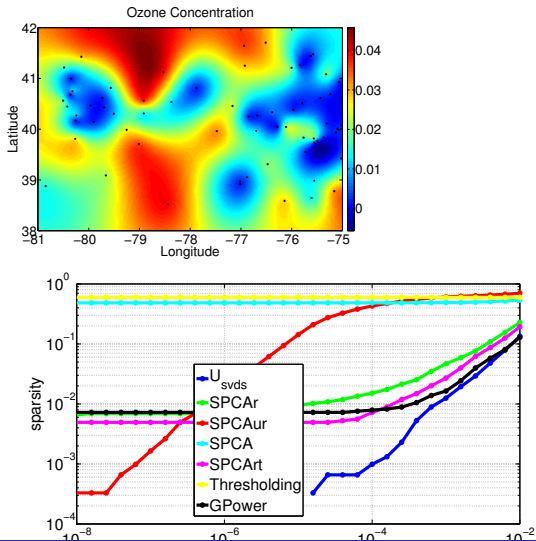
Figure : Sparsity of the subspace estimate from complete observations given by six algorithms.

Sparse PCA

Metrics	SPCAr	SPCAur	SPCA	SPCArt	Thres	GPower
Subspace Residual	6.0480e-5	0.0020	0.0818	0.0034	0.4680	0.0300
Runtime	0.0813	0.2442	0.0625	0.0859	0.0005	0.0797
Minimum Angle	90.0000	90.0000	87.3738	89.9093	83.7886	86.9081
Sparsity (10^{-4})	0.7697	0.8337	0.3298	0.8210	0.5498	0.1876
Convergence Proportion	0.2764	0.9975	1.0000	1.0000	1.0000	1.0000

Table : Subspace residuals, runtimes, minimum angles, sparsities when $\alpha = 10^{-4}$ and convergence proportions of these six algorithms including SPCA [Zou et al., 2006], SPCArt [Hu et al., 2014], and GPower [Journée et al., 2010].

Sparse PCA Results

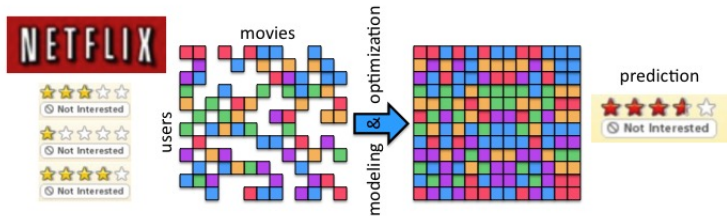


On our ozone data, SPCAur gives the best sparsity result while still maintaining orthogonality, while SPCA and Thresholding do not.

Calibration SVD

$$\begin{aligned} & \underset{X \in \mathbb{R}^{n \times m}, g}{\text{minimize}} && \|P_{\Psi}(g(X - M))\|_F^2 && (10) \\ & \text{subject to} && g \text{ is } L\text{-Lipschitz and monotonic} \\ & && X \text{ low-rank} \end{aligned}$$

Example 1: Recommender Systems

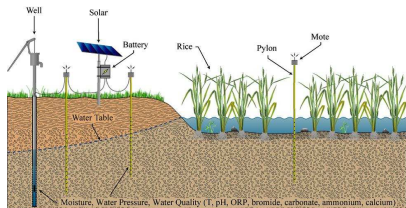
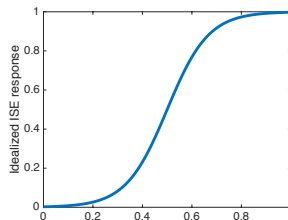


Example 2: Blind Sensor Calibration



Example 2: Blind Sensor Calibration

Ion Selective Electrodes have a nonlinear response to their ions (pH, ammonium, calcium, etc)



Single Index Model

Suppose we have predictor variables x and response variables y , and we seek a transformation g and vector w relating the two such that

$$\mathbb{E}[y|x] = g\left(x^T w\right).$$

- Generalized Linear Model: g is known, $y|x$ are RVs from an exponential family distribution parameterized by w .
 - Includes linear regression, log-linear regression, and logistic regression
- Single Index Model: Both g and w are unknown.

Single Index Model Learning

We seek a transformation g and vector w such that

$$\mathbb{E}[y|x] = g(x^T w) .$$

Theorem (Kalai et al 2009, Kakade et al 2011)

Suppose $(x_i, y_i) \in \mathcal{B}_n \times [0, 1]$, $i = 1, \dots, p$ are draws from a distribution where $\mathbb{E}[y|x] = g(x^T w)$ for monotonic G -Lipschitz g and $\|w\| \leq 1$. There is a $\text{poly}(1/\epsilon, \log(1/\delta), n)$ time algorithm that, given any $\delta, \epsilon > 0$, with probability $\geq 1 - \delta$ outputs $h(x) = \hat{g}(\hat{w}^T x)$ with

$$\text{err}(h) = \mathbb{E}_{y|x} [(g(x^T w) - h(x))^2] < \epsilon$$

Single Index Model Learning

Algorithm 3 Lipschitz-Isotron Algorithm, Kakade et al 2011

Given $T > 0$, $(x_i, y_i)_{i=1}^P$;

Set $w^{(1)} := 1$;

for $t = 1, 2, \dots, T$ **do**

Update g using Lipschitz-PAV: $g^{(t)} = LPAV((x_i^T w^{(t)}, y_i)_{i=1}^P)$.

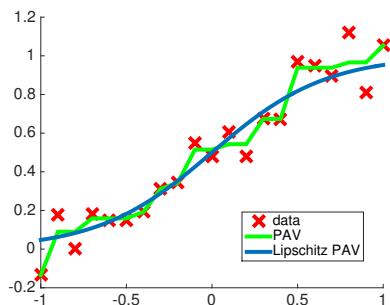
Update w using gradient descent:

$$w^{(t+1)} = w^{(t)} + \frac{1}{P} \sum_{i=1}^P \left(y_i - g^{(t)}(x_i^T w^{(t)}) \right) x_i$$

end for

Lipschitz Pool Adjacent Violator

- The Pool Adjacent Violator (PAV) algorithm pools points and averages to minimize mean squared error $g(x_i) - y_i$. PAV
- L-PAV adds the additional constraint of a given Lipschitz constant.



High-rank Matrices

For Z low-rank,

$$Y_{ij} = g(Z_{ij}) = \frac{1}{1 + \exp^{-\gamma Z_{ij}}}, Y \text{ has full rank.}$$

$$Y_{ij} = g(Z_{ij}) = \text{quantize_to_grid}(Z_{ij}), Y \text{ has full rank.}$$

High-rank Matrices: Effective rank

These matrices even have high effective rank.

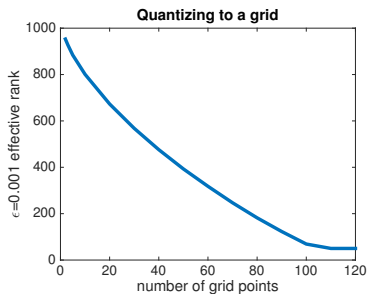
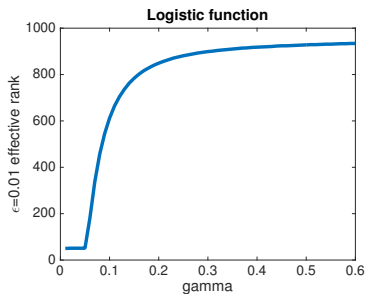
Definition

The **effective rank** of an $n \times m$ matrix Y , $m < n$, with singular values σ_j is

$$r_\epsilon(Y) = \min \left\{ k \in \mathbb{N} : \sqrt{\frac{\sum_{j=k+1}^m \sigma_j^2}{\sum_{j=1}^m \sigma_j^2}} \leq \epsilon \right\} .$$

High-rank Matrices: Effective rank

These matrices even have high effective rank.
For a rank-50, 1000x1000 matrix:



Problem Formulation

Our model is as follows:

- **Low-rank matrix** $Z^* \in \mathbb{R}^{n \times m}$ with $m \leq n$ and (for now, known) rank $r \ll m$.
- **Lipschitz link function** $g^* : \mathbb{R} \rightarrow \mathbb{R}$, monotonic, Lipschitz
- **Noise matrix** $N \in \mathbb{R}^{n \times m}$ with iid entries $\mathbb{E}[N] = 0$.
- **Samples of matrix entries** $\Psi \in \{1, \dots, n\} \times \{1, \dots, m\}$ is a multiset, sampled independently with replacement.

We observe $Y_{ij} = g^*(Z_{ij}^*) + N_{ij}$ for $(i, j) \in \Psi$

and we wish to recover g^*, Z^* .

Optimization Formulation

$$\min_{g, Z} \sum_{\psi} (g(Z_{i,j}) - Y_{i,j})^2$$

subj. to $g : \mathbb{R} \rightarrow \mathbb{R}$ is Lipschitz and monotone
 $\text{rank}(Z) \leq r$

Non-convex in each variable, but we can alternate the standard approaches:

- Use gradient descent and projection onto the low-rank cone for Z .
- Use LPAV for g .

We call this algorithm MMC-LS.

MMC-LS Algorithm

Algorithm 4 MMC-LS

Given max iterations $T > 0$, step size $\eta > 0$, rank r , data Y_Ψ

Init $\hat{g}^{(0)}(z) = \frac{|\Psi|}{mn}z$, $\hat{Z}^{(0)} = \frac{mn}{|\Psi|}Y_0$, where Y_0 zero-filled Y_Ψ .

for $t = 1, 2, \dots, T$ **do**

Update \hat{Z} using gradient descent:

$$\hat{Z}_{i,j}^{(t)} = \hat{Z}_{i,j}^{(t-1)} - \eta \left(\hat{g}^{t-1} \left(\hat{Z}_{i,j}^{(t-1)} \right) - Y_{i,j} \right) \left(\hat{g}^{t-1} \right)' \left(\hat{Z}_{i,j}^{(t-1)} \right) \mathbb{I}_{(i,j) \in \Psi}$$

Project: $\hat{Z}^{(t)} = \mathcal{P}_r(\hat{Z}^{(t)})$

Update \hat{g} : $\hat{g}^{(t)} = LPAV \left(\{ (\hat{Z}_{i,j}^{(t)}, Y_{i,j}) \text{ for } (i,j) \in \Psi \} \right)$.

end for

Optimization of Calibrated Loss

Let $\Phi : \mathbb{R} \rightarrow \mathbb{R}$ be a differentiable function that satisfies $\Phi' = g^*$. Since g^* is monotonic, Φ is convex. Consider:

$$L(\Phi, Z) = \sum_{(i,j) \in \Psi} \Phi(Z_{i,j}) - Y_{i,j} Z_{i,j}$$

Differentiating with respect to Z we get that a minimizer satisfies $\sum_{(i,j) \in \Psi} g^*(Z_{i,j}) - Y_{i,j} = 0$; in other words, Z^* is a minimizer in expectation. So $L(\Phi, Z)$ is a calibrated loss for our problem.

MMC-c Algorithm

Algorithm 5 MMC-calibrated

Given max iterations $T > 0$, step size $\eta > 0$, rank r , data Y_Ψ

Init $\hat{g}^{(0)}(z) = \frac{|\Psi|}{mn}z$, $\hat{Z}^{(0)} = \frac{mn}{|\Psi|}Y_0$, where Y_0 zero-filled Y_Ψ .

for $t = 1, 2, \dots, T$ **do**

Update \hat{Z} using gradient descent:

$$\hat{Z}_{i,j}^{(t)} = \hat{Z}_{i,j}^{(t-1)} - \eta \left(\hat{g}^{t-1} \left(\hat{Z}_{i,j}^{(t-1)} \right) - Y_{i,j} \right) \mathbb{I}_{(i,j) \in \Psi}$$

Project: $\hat{Z}^{(t)} = \mathcal{P}_r(\hat{Z}^{(t)})$

Update g : $g^{(t)} = \text{LPAV} \left(\{(\hat{Z}_{i,j}^{(t)}, Y_{i,j}) \text{ for } (i,j) \in \Psi\} \right)$.

end for

Remarks

MMC consists of three steps: gradient descent, projection, and LPAV.

- The gradient descent step requires a step size parameter η ; we chose a small constant stepsize by cross validation.
- The projection requires rank r . For our implementation, we started with a small r and increased it, in the same vein as Wen, Yin, and Zhang 2012.
- LPAV is the solution of a QP. Ravi developed an ADMM implementation as well.

MSE Analysis of MMC-c

Let $\hat{M} = \hat{g}(\hat{Z})$ and $M^* = g^*(Z^*)$.

Define the MSE as

$$MSE(\hat{M}) = \mathbb{E} \left[\frac{1}{mn} \sum_{i=1}^n \sum_{j=1}^m \left(\hat{M}_{i,j} - M_{i,j}^* \right)^2 \right]$$

MSE Analysis of MMC-c

Theorem (MSE of MMC-c after one iteration, Ganti, Balzano, Willett 2015)

Let $\|Z^*\| = O(\sqrt{n})$ and $\sigma_{r+1}(Y) = \tilde{O}(\sqrt{n})$ with high probability. Let $\alpha = \|M^* - Z^*\|$. Furthermore, assume that elements of Z^* and Y are bounded in absolute value by 1.

Then the MSE of one step of MMC ($T = 1$) is bounded by

$$\text{MSE}(\hat{M}) \leq O\left(\sqrt{\frac{r}{m}} + \frac{mn}{|\Psi|^{3/2}} + \sqrt{\frac{r\alpha}{m\sqrt{n}}\left(1 + \frac{\alpha}{\sqrt{n}}\right)}\right).$$

MSE Analysis of MMC-c

Theorem (MSE of MMC-c after one iteration, Ganti, Balzano, Willett 2015)

In addition to the previous assumptions, let

$$\alpha = \|M^* - Z^*\| = O(\sqrt{n}) .$$

Then the MSE of one step of MMC is bounded by

$$\text{MSE}(\hat{M}) \leq O\left(\sqrt{\frac{r}{m}} + \frac{mn}{|\Psi|^{3/2}}\right) .$$

Synthetic Data

Z^* is 30×20 and rank 5.

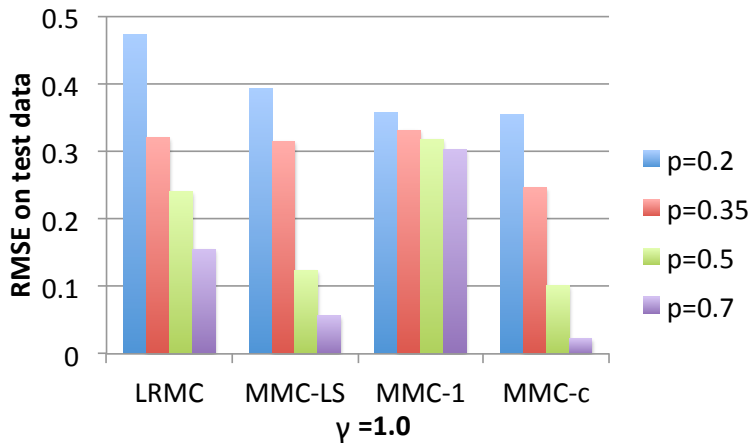
$N = 0$

Toy ISE calibration function: $g^*(z) = 1/(1 + \exp^{-\gamma z})$

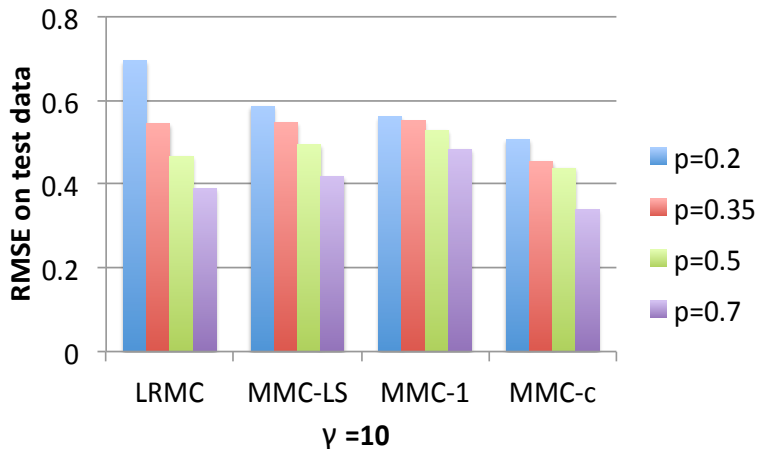
Vary $\gamma = 1, 10, 40$.

Vary probability of observation $p = .2, .35, .5, .7$.

Synthetic Data

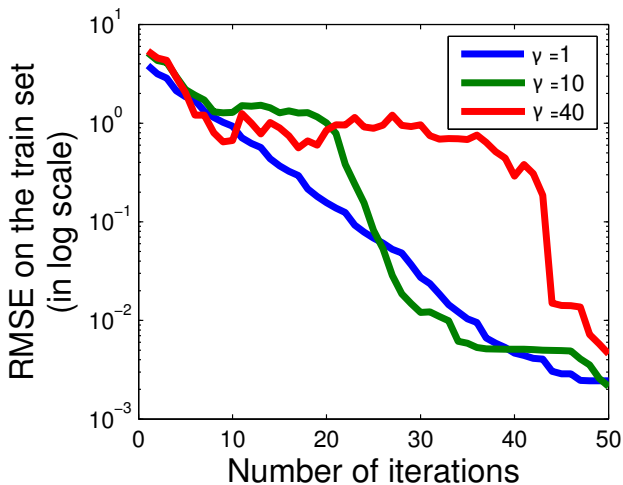


Synthetic Data



$\gamma = 40$

Synthetic Data



Real Data

- Paper recommendation: 3426 features from 50 scholars' research profiles.
- Jester: 4.1 Million continuous ratings (-10.00 to +10.00) of 100 jokes from 73,421 users.
- Movie lens: 100,000 ratings from 1000 users on 1700 movies.
- Cameraman: Dictionary learning on patches of the image.

Dataset	Dimension	$ \Psi $	$r_{0.01}(Y)$
PaperReco	3426×50	34294 (20%)	47
Jester-3	24938×100	124690 (5%)	66
ML-100k	1682×943	64000 (4%)	391
Cameraman	1536×512	157016 (20%)	393

Real Data Performance

RMSE on a held-out test set:

Dataset	$ \Psi /mn$	LMaFit-A	MMC-c $T = 1$	MMC-c
PaperReco	20%	0.4026	0.4247	0.2965
Jester-3	5%	6.8728	5.327	5.2348
ML-100k	4%	3.3101	1.388	1.1533
Cameraman	20%	0.0754	0.1656	0.06885

Conclusion

- Low-rank matrix constraints may be defined by a smooth manifold.
- Standard manifold optimization methods when applied to these non-convex problems work well.
- The GROUSE algorithm is just the natural incremental gradient on the Grassmannian for subspace learning.
- GROUSE is equivalent to a missing-data ISVD and it exhibits global convergence behavior.
- Adding sparsity regularizers works empirically but there is much to understand theoretically!

Thank you! Questions?



Balzano, L., Nowak, R., and Recht, B. (2010).

Online identification and tracking of subspaces from highly incomplete information.
In Proceedings of the Allerton conference on Communication, Control, and Computing.



Balzano, L. and Wright, S. J. (2013).

On GROUSE and incremental SVD.
In IEEE Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP).



Balzano, L. and Wright, S. J. (2015).

Local convergence of an algorithm for subspace identification from partial data.
Foundations of Computational Mathematics, 15(5):1279–1314.



Ding, Q. and Kolaczyk, E. D. (2013).

A compressed PCA subspace method for anomaly detection in high-dimensional data.
IEEE Transactions on Information Theory, 59(11).



Eckart, C. and Young, G. (1936).

The approximation of one matrix by another of lower rank.
Psychometrika, 1(3):211–218.



Edelman, A., Arias, T. A., and Smith, S. T. (1998).

The geometry of algorithms with orthogonality constraints.
SIAM Journal on Matrix Analysis and Applications, 20(2):303–353.



He, J., Balzano, L., and Szelam, A. (2012).

Incremental gradient on the grassmannian for online foreground and background separation in subsampled video.

In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.



Hu, Z., Pan, G., Wang, Y., and Wu, Z. (2014).

Sparse principal component analysis via rotation and truncation.



Journée, M., Nesterov, Y., Richtárik, P., and Sepulchre, R. (2010).

Generalized power method for sparse principal component analysis.

The Journal of Machine Learning Research, 11:517–553.



Kennedy, R., Balzano, L., Wright, S. J., and Taylor, C. J. (2016).

Online algorithms for factorization-based structure from motion.

Computer Vision and Image Understanding.



Schmidt, E. (1907).

Zur theorie der linearen und nicht linearen integralgleichungen. i teil. entwicklung willkürlichen funktionen nach system vorgeschriebener.

Mathematische Annalen, 63:433–476.



Stewart, G. (2011).

Fredholm, hilbert, schmidt: Three fundamental papers on integral equations.

Available at www.cs.umd.edu/~stewart/FHS.pdf.



Stewart, G. W. (1993).

On the early history of the singular value decomposition.

SIAM review, 35(4):551–566.



Zhang, D. and Balzano, L. (2016).

Global convergence of a grassmannian gradient descent algorithm for subspace estimation.



Zou, H., Hastie, T., and Tibshirani, R. (2006).

Sparse principal component analysis.

Journal of computational and graphical statistics, 15(2):265–286.

GROUSE Comments

The GROUSE update is essentially a projection of a step along the search direction $r_t w_t^T$. Defining the inconsistency measure

$$\mathcal{E}(U_t) := \min_{w_t} \|[U_t]_{\Psi_t} w_t - [v_t]_{\Psi_t}\|_2^2,$$

we have

$$\frac{d\mathcal{E}}{dU_t} = -2r_t w_t^T,$$

so we see that the GROUSE search direction is the negative gradient of \mathcal{E} .

The GROUSE update has much in common with quasi-Newton updates in optimization, in that it makes the **minimal adjustment required to match the latest observations**, while retaining a certain desired structure.

Partial-data ISVD and GROUSE

This ISVD and GROUSE seem similar:

- We defined them both to compute and use w_t to extract the missing information from U_t and $[v_t]_{\Psi_t}$.
- Both generate a sequence $\{U_t\}$ of estimates of \mathcal{S} .
- Both use only U_t and $[v_t]_{\Psi_t}$ to generate U_{t+1} .
- Neither has different confidence for different subspaces of the target subspace \mathcal{S} ; both maintain a “flat” approximation.

Indeed, we can show that ISVD and GROUSE are **identical** for a certain choice of the step-size parameter η_t .

The choice of η_t is *not* the same as the “optimal” choice in GROUSE, but it works fairly well in practice.

Relating partial-data ISVD and GROUSE

Theorem

Suppose we have the same U_t and $[v_t]_{\Psi_t}$ at the t^{th} iterations of ISVD and GROUSE. Then we can construct an $\eta_t > 0$ in GROUSE such that the next iterates U_{t+1} of both algorithms are identical, to within an orthogonal transformation by the $d \times d$ matrix

$$W_t := \begin{bmatrix} \frac{w_t}{\|w_t\|} & Z_t \end{bmatrix},$$

where Z_t is a $d \times (d - 1)$ matrix whose orthonormal columns span the nullspace of w_t^T .

GROUSE and ISVD: Details

The precise values for which GROUSE and ISVD are identical are:

$$\lambda = \frac{1}{2} \left[(\|w_t\|^2 + \|r_t\|^2 + 1) + \sqrt{(\|w_t\|^2 + \|r_t\|^2 + 1)^2 - 4\|r_t\|^2} \right]$$

This is the first eigenvalue of the matrix $\begin{bmatrix} \mathcal{I}_d & w_t \\ 0 & \|r_t\| \end{bmatrix}$;

the next $d - 1$ eigenvalues are 1 by the interleaving theorem.

$$\beta = \frac{\|r_t\|^2 \|w_t\|^2}{\|r_t\|^2 \|w_t\|^2 + (\lambda - \|r_t\|^2)^2} \quad ; \quad \eta_t = \frac{1}{\sigma_t} \arcsin \beta.$$

Incremental SVD with Missing Data Options

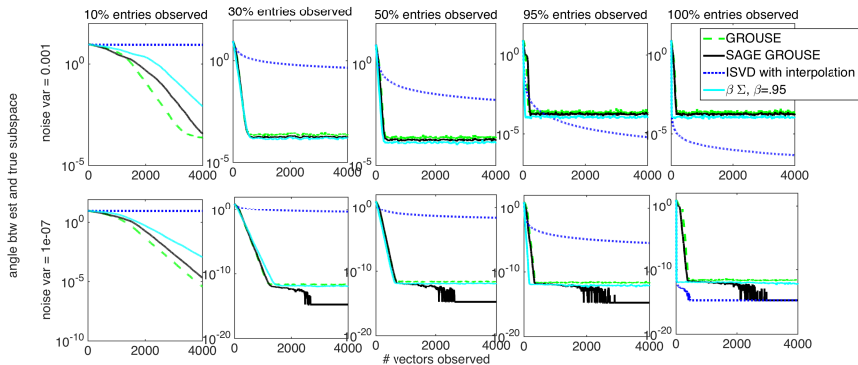
projection weights $w = \arg \min_a \|P_{\Psi_t}(U_t a - v_t)\|_2^2$;
 residual: $r_t = v_t - U w$ on Ψ_t ; zero otherwise.

$$\text{ISVD with interpolation: } \begin{bmatrix} U & \frac{r_t}{\|r_t\|} \end{bmatrix} \begin{bmatrix} \Sigma & w \\ 0 & \|r_t\| \end{bmatrix} \begin{bmatrix} V & 0 \\ 0 & 1 \end{bmatrix}^T$$

$$\text{SAGE GROUSE: } \begin{bmatrix} U & \frac{r_t}{\|r_t\|} \end{bmatrix} \begin{bmatrix} \mathcal{I}_d & w \\ 0 & \|r_t\| \end{bmatrix} \begin{bmatrix} V & 0 \\ 0 & 1 \end{bmatrix}^T$$

$$\text{Brand Algorithm } (\beta \leq 1): \begin{bmatrix} U & \frac{r_t}{\|r_t\|} \end{bmatrix} \begin{bmatrix} \beta \Sigma & w \\ 0 & \|r_t\| \end{bmatrix} \begin{bmatrix} V & 0 \\ 0 & 1 \end{bmatrix}^T$$

Incremental SVD with Missing Data Performance



isvd grouse