Learning Linear Dynamical Systems from Multivariate Time Series: A Matrix Factorization Based Framework - Supplemental Material

Zitao Liu* Milos Hauskrecht*

Notations

We use the following notations in this supplemental material. The notation is consistent with the original paper.

- $\hat{\mathbf{z}}_{t|t-1} = \mathbb{E}[\mathbf{z}_t|\{\mathbf{y}\}_1^{t-1}]$ is the *priori* estimation
- $\hat{\mathbf{z}}_{t-1|t-1} = \mathbb{E}[\mathbf{z}_{t-1}|\{\mathbf{y}\}_1^{t-1}]$ is the *posteriori* estimation.
- $P_{t|t-1} = \mathbb{E}[(\mathbf{z}_t - \hat{\mathbf{z}}_{t|t-1})(\mathbf{z}_t - \hat{\mathbf{z}}_{t|t-1})^{'}]$ is the *priori* estimate error covariance.
- $P_{t-1|t-1} = \mathbb{E}[(\mathbf{z}_{t-1} - \hat{\mathbf{z}}_{t-1|t-1})(\mathbf{z}_{t-1} - \hat{\mathbf{z}}_{t-1|t-1})^{'}]$ is the *posteriori* estimate error covariance.
- $\hat{\mathbf{z}}_{t|T} \equiv \mathbb{E}[\mathbf{z}_t|\mathbf{y}]$, $M_{t|T} \equiv \mathbb{E}[\mathbf{z}_t\mathbf{z}_t^{'}|\mathbf{y}]$, $M_{t,t-1|T} \equiv \mathbb{E}[\mathbf{z}_t\mathbf{z}_{t-1}^{'}|\mathbf{y}]$, $P_{t|T} = \mathbb{VAR}[\mathbf{z}_t|\mathbf{y}]$, and $P_{t,t-1|T} = \mathbb{VAR}[\mathbf{z}_t\mathbf{z}_{t-1}^{'}|\mathbf{y}]$

1 Kalman Filter Algorithm

The details of Kalman filter algorithm are shown in Algorithm 1.

Algorithm 1 Kalman filter algorithm for LDS

INPUT: Current step LDS parameters: $\Omega = \{A, C, Q, R, \boldsymbol{\xi}, \Psi\}$.

PROCEDURE:

// Initialize the recursion
$\hat{\mathbf{z}}_{1|1} = \boldsymbol{\xi}$ and $P_{1|1} = \Psi$.
// Start the recursion
for t = 2 → T **do**
  // Time Update:
  $\hat{\mathbf{z}}_{t|t-1} = A\hat{\mathbf{z}}_{t-1|t-1}$
  $P_{t|t-1} = AP_{t-1|t-1}A^{'} + Q$
  // Measure Update:
  $K_t = P_{t|t-1}C^{'}(CP_{t|t-1}C^{'} + R)^{-1}$
  $\hat{\mathbf{z}}_{t|t} = \hat{\mathbf{z}}_{t|t-1} + K_t(\mathbf{y}_t - C\hat{\mathbf{z}}_{t|t-1})$
  $P_{t|t} = P_{t|t-1} - K_tCP_{t|t-1}$
end for

OUTPUT: $\{\hat{\mathbf{z}}_{t|t-1}\}_{t=2}^T$, $\{\hat{\mathbf{z}}_{t|t}\}_{t=1}^T$, $\{P_{t|t}\}_{t=1}^T$, $\{P_{t|t-1}\}_{t=2}^T$ and $\{K_t\}_{t=1}^T$.

*Computer Science Department, University of Pittsburgh, Pittsburgh, PA USA. Email: {ztliu, milos}@cs.pitt.edu

2 Kalman Smoothing Algorithm

The details of Kalman smoothing algorithm are shown in Algorithm 2.

Algorithm 2 EM: E-step Smoothing algorithm for LDS

INPUT:
- Output from Kalman filter algorithm: $\{\hat{\mathbf{z}}_{t|t-1}\}_{t=2}^T$, $\{\hat{\mathbf{z}}_{t|t}\}_{t=1}^T$, $\{P_{t|t}\}_{t=1}^T$, $\{P_{t|t-1}\}_{t=2}^T$ and $\{K_t\}_{t=1}^T$.
- Current step LDS parameters: $\Omega = \{A, C, Q, R, \boldsymbol{\xi}, \Psi\}$.

PROCEDURE:

// Initialize the recursion
$M_{T|T} = P_{T|T} + \hat{\mathbf{z}}_{T|T}\hat{\mathbf{z}}_{T|T}^{'}$
$J_{T-1} = P_{T-1|T-1}A^{'}(P_{T|T-1})^{-1}$
$P_{T-1|T} = P_{T-1|T-1} + J_{T-1}(P_{T|T} - P_{T|T-1})J_{T-1}^{'}$
$\hat{\mathbf{z}}_{T-1|T} = \hat{\mathbf{z}}_{T-1|T-1} + J_{T-1}(\hat{\mathbf{z}}_{T|T} - A\hat{\mathbf{z}}_{T-1|T-1})$
$P_{T,T-1|T} = (I - K_TC)AP_{T-1|T-1}$
$M_{T,T-1|T} = P_{T,T-1|T} + \hat{\mathbf{z}}_{T|T}\hat{\mathbf{z}}_{T-1|T}^{'}$
// Start the recursion
for t = T-1 → 1 **do**
  $M_{t|T} = P_{t|T} + \hat{\mathbf{z}}_{t|T}\hat{\mathbf{z}}_{t|T}^{'}$
  $J_{t-1} = P_{t-1|t-1}A^{'}(P_{t|t-1})^{-1}$
  $P_{t,t-1|T} = P_{t|t}J_{t-1}^{'} + J_t(P_{t+1,t|T} - AP_{t|t})J_{t-1}^{'}$
  $M_{t,t-1|T} = P_{t,t-1|T} + \hat{\mathbf{z}}_{t|T}\hat{\mathbf{z}}_{t-1|T}^{'}$
  $\hat{\mathbf{z}}_{t-1|T} = \hat{\mathbf{z}}_{t-1|t-1} + J_{t-1}(\hat{\mathbf{z}}_{t|T} - A\hat{\mathbf{z}}_{t-1|t-1})$
  $P_{t-1|T} = P_{t-1|t-1} + J_{t-1}(P_{t|T} - P_{t|t-1})J_{t-1}^{'}$
end for

OUTPUT: $\{\hat{\mathbf{z}}_{t-1|T}\}_{t=1}^T$, $\{M_{t|T}\}_{t=1}^T$ and $\{M_{t,t-1|T}\}_{t=1}^T$.

3 Theorem Proof

3.1 Theorem 3.1 Proof

THEOREM 3.1. *Generalized gradient descent with a fixed step size $\rho \leq 1/2(\|\mathbf{Z}_-\mathbf{Z}_-^\top\|_F + \gamma/\lambda)$ for minimizing eq.(3.22) has convergence rate $O(1/k)$, where* k *is the number of iterations.*

Proof. $g(A)$ is differentiable with respect to A, and its gradient is

$$\bigtriangledown g(A) = 2(A\mathbf{Z}_-\mathbf{Z}_-^\top - \mathbf{Z}_+\mathbf{Z}_-^\top + \gamma/\lambda A)$$

Using simple algebraic manipulation we arrive at

$$
\begin{aligned}
& \|\nabla g(X) - \nabla g(Y)\|_F \\
=& 2\|(X-Y)(\mathbf{Z}_-\mathbf{Z}_-^\top) + \gamma/\lambda(X-Y)\|_F \\
\leq& 2\|\mathbf{Z}_-\mathbf{Z}_-^\top\|_F \cdot \|X-Y\|_F + 2\gamma/\lambda \cdot \|X-Y\|_F \\
=& 2(\|\mathbf{Z}_-\mathbf{Z}_-^\top\|_F + \gamma/\lambda) \cdot \|X-Y\|_F
\end{aligned}
$$

The inequality holds because of the sub-multiplicative property of Frobenius norm. Since we know for eq.(3.22), $\min_A g(A) + \gamma_A\|A\|_*$, and $g(A)$ has Lipschitz continuous gradient with constant $2(\|\mathbf{Z}_-\mathbf{Z}_-^\top\|_F + \gamma/\lambda)$, according to [1, 2] we have

$$
\begin{aligned}
& \left\| g(A^{(k)}) + \gamma_A\|A^{(k)}\|_* - g(A^{(*)}) - \gamma_A\|A^{(*)}\|_* \right\| \\
\leq& \left\| A^{(0)} - A^* \right\|_F^2 / 2tk
\end{aligned}
$$

where $A^{(0)}$ is the initial value and A^* is the optimal value for A; k is the number of iterations. ■

3.2 Theorem 3.2 Proof

THEOREM 3.2. *Minimizing A from eq.(3.7) with $\mathcal{R}_A(A) = \emptyset$ is equivalent to minimizing the following problem:*

$$
\min_a a^\top Ba - 2q^\top a \tag{3.1}
$$

where $a = \mathrm{vec}(A^\top)$, $B = I_d \otimes (\mathbf{Z}_-\mathbf{Z}_-^\top)$, $q = (I_d \otimes \mathbf{Z}_-\mathbf{Z}_+^\top)\,\mathrm{vec}(I_d)$.

Proof. We will use the following equation to show the equivalence.

$$
tr(A_{k\times l}B_{l\times m}C_{m\times n}) = \mathrm{vec}(A^\top)^\top(I_k \otimes B)\,\mathrm{vec}(C)
$$

$$
\begin{aligned}
& \min_A \|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 \\
\Leftrightarrow & \min_A Tr[(\mathbf{Z}_+^\top - \mathbf{Z}_-^\top A^\top)(\mathbf{Z}_+ - A\mathbf{Z}_-)] \\
\Leftrightarrow & \min_A Tr[A\mathbf{Z}_-\mathbf{Z}_-^\top A^\top - 2I_d\mathbf{Z}_+\mathbf{Z}_-^\top A^\top] \\
\Leftrightarrow & \min_A \mathrm{vec}(A^\top)^\top(I_d \otimes \mathbf{Z}_-\mathbf{Z}_-^\top)\,\mathrm{vec}(A^\top) \\
& - 2\,\mathrm{vec}(I_d)^\top(I_d \otimes \mathbf{Z}_+\mathbf{Z}_-^\top)\,\mathrm{vec}(A^\top) \\
\Leftrightarrow & \min_a a^\top(I_d \otimes \mathbf{Z}_-\mathbf{Z}_-^\top)a - 2\,\mathrm{vec}(I_d)^\top(I_d \otimes \mathbf{Z}_+\mathbf{Z}_-^\top)a \\
\Leftrightarrow & \min_a a^\top(I_d \otimes \mathbf{Z}_-\mathbf{Z}_-^\top)a - 2\Big((I_d \otimes \mathbf{Z}_-\mathbf{Z}_+^\top)\,\mathrm{vec}(I_d)\Big)^\top a \\
\Leftrightarrow & \min_a a^\top Ba - 2q^\top a
\end{aligned}
$$

where $a = \mathrm{vec}(A^\top)$, $B = I_d \otimes \mathbf{Z}_-\mathbf{Z}_-^\top$ and $q = (I_d \otimes \mathbf{Z}_-\mathbf{Z}_+^\top)\,\mathrm{vec}(I_d)$. ■

4 Qualitative Prediction Analysis

In this section, we qualitatively show the prediction effectiveness of the gLDS-smooth model from our framework. Figure 1 and Figure 2 show the predictions results for the flour price series in Minneapolis and Kansas City.

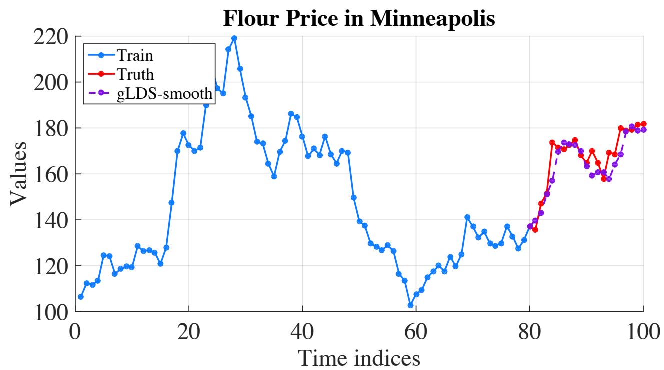

Figure 1: Predictions for flour price series in Minneapolis.

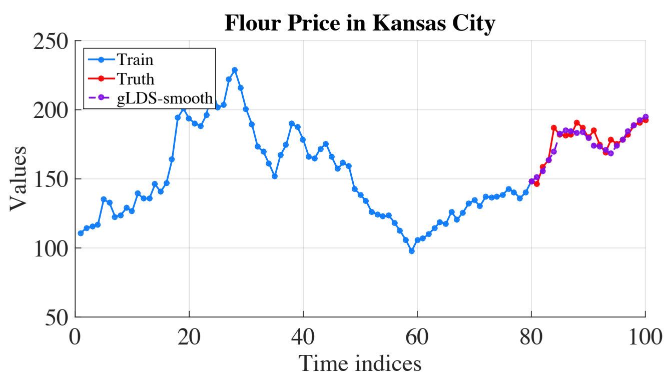

Figure 2: Predictions for flour price series in Kansas City.

4.1 Quantitative Prediction Analysis

In this section, we quantitatively compute and compare the prediction accuracy of the proposed methods (gLDS-ridge and gLDS-smooth) with the standard LDS learning approaches: EM and spectral algorithms. The results are shown in Table 1 and Table 2.

5 Stability Effects of gLDS-stable

In this section, we show the stability effects of the gLDS-stable model learned using our framework by generating the simulated sequences in the future for *flourprice*, *h20_evap* and *clinical* datasets, which are shown in Figures 3 - 5.

Table 1: Average-MAPE results on *flourprice* dataset.

	Training: 80%		**Training: 90%**	
# of states	5	10	5	10
Spectral	6.25	5.86	6.61	5.93
EM	3.62	4.15	3.63	3.94
gLDS-ridge	3.37	3.14	3.29	2.82
gLDS-smooth	**3.24**	**2.71**	**2.86**	**2.50**

Table 2: Average-MAPE results on *h2o_evap* dataset.

	Training: 80%		**Training: 90%**	
# of states	5	10	5	10
Spectral	36.26	32.20	13.73	15.88
EM	39.53	68.68	17.33	17.46
gLDS-ridge	27.97	28.53	16.12	14.42
gLDS-smooth	**26.38**	**26.46**	**14.01**	**14.08**

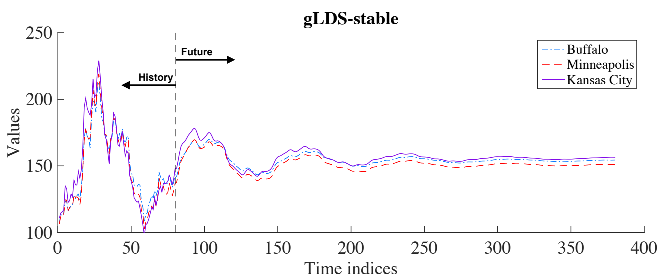

Figure 3: Training data and simulated sequences from gLDS-stable model in *fourprice* data.

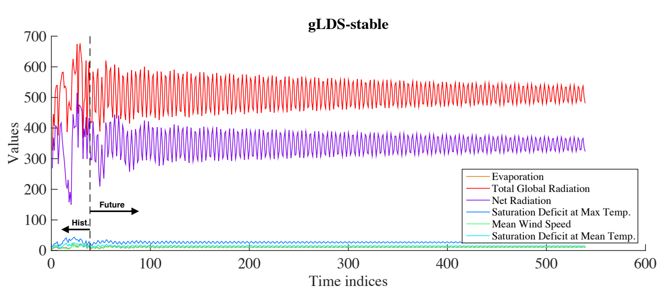

Figure 4: Training data and simulated sequences from gLDS-stable model in *h2o_evap* data.

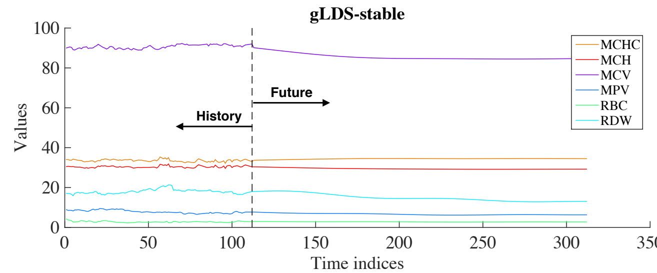

Figure 5: Training data and simulated sequences from gLDS-stable model in *clinical* data for one patient.

6 Sparsification Effects of gLDS-low-rank

In this section, we show the sparsification effects of the gLDS-low-rank model learned using our framework. The gLDS-low-rank model is able to identify the intrinsic dimensionality of the hidden state space. The results are shown in Figure 6 and Figure 7.

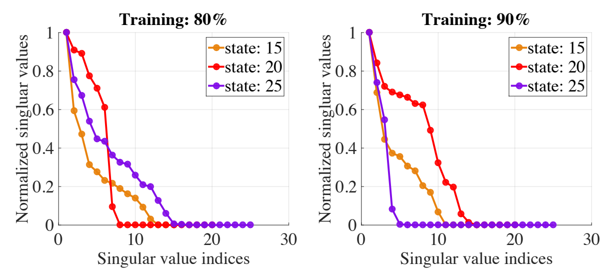

Figure 6: Intrinsic dimensionality recovery of the hidden state space in *flourprice* dataset.

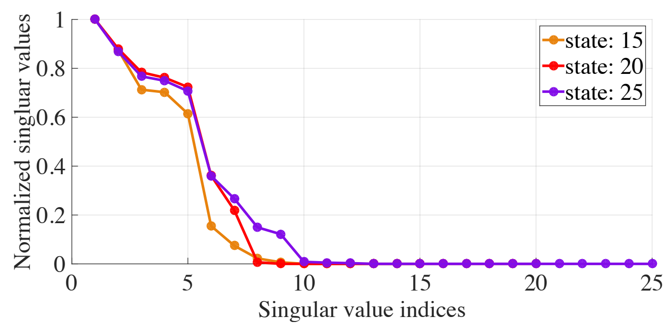

Figure 7: Intrinsic dimensionality recovery of the hidden state space in *clinical* dataset.

References

[1] M. Fornasier and H. Rauhut, *Iterative thresholding algorithms*, Applied and Computational Harmonic Analysis, 25 (2008), pp. 187–208.

[2] N. Z. Shor, *The rate of convergence of the generalized gradient descent method*, Cybernetics and Systems Analysis, 4 (1968), pp. 79–80.