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

Zitao Liu   Milos Hauskrecht

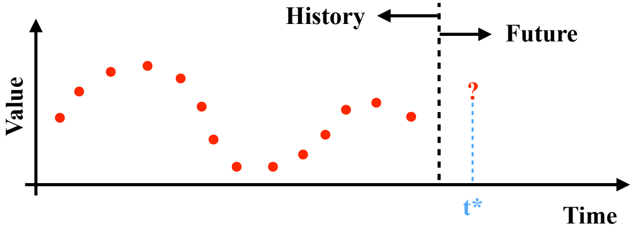
University of Pittsburgh

ztliu@cs.pitt.edu

ztliu@cs.pitt.edu

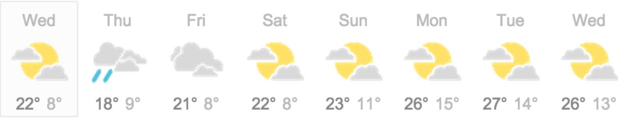
Problem

Make future predictions for time series.



Problem

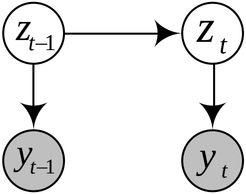
Make future predictions for time series.



Linear Dynamical System (LDS)

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + \boldsymbol{\epsilon}_t \qquad \mathbf{y}_t = C\mathbf{z}_t + \boldsymbol{\zeta}_t$$

$\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, Q)$, $\boldsymbol{\zeta}_t \sim \mathcal{N}(0, R)$ and $z_1 \sim \mathcal{N}(\boldsymbol{\xi}, \Psi)$.



- $\{\mathbf{y}_t\}$: time series of observations
- $\{\mathbf{z}_t\}$: hidden states driving the dynamics
- Parameters $\Lambda = \{A, C, Q, R, \boldsymbol{\xi}, \Psi\}$
- Also known as Kalman filter [Kalman, 1960]

Advantages of Linear Dynamical System (LDS)

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + \epsilon_t \qquad \mathbf{y}_t = C\mathbf{z}_t + \zeta_t$$

$$\epsilon_t \sim \mathcal{N}(0, Q),\ \zeta_t \sim \mathcal{N}(0, R) \text{ and } z_1 \sim \mathcal{N}(\xi, \Psi).$$

Advantages:

- A multivariate model
- Efficiently exact inference and predictions

Learning LDS

Well studied learning algorithms:

- EM algorithms: $\mathcal{Q} = \mathbb{E}_{\mathbf{z}}\Big[\log p(\mathbf{z}, \mathbf{y})\Big]$
- Spectral methods: hankel matrix + SVD

Why? Drive the dynamics to behave what we expect.
Regularization, stability, etc.

- Constraints on LDS Inference
- Constraints on LDS Learning ($\bigstar$)

Question: How can we add constraints in the learning process?

Learning LDS via Matrix Factorization

Well studied learning algorithms:

- EM algorithms: $\mathcal{Q} = \mathbb{E}_{\mathbf{z}}\Big[\log p(\mathbf{z}, \mathbf{y})\Big]$
- Spectral methods: hankel matrix + SVD

Our approach:

Learning (Constrained) LDS via Matrix Factorization!

Notation

Given a collection of N multivariate time series sequences $\{\mathbf{Y}^1, \mathbf{Y}^2, \cdots, \mathbf{Y}^N\}$,

- $\mathbf{Y}^m = [\mathbf{y}_1^m, \cdots, \mathbf{y}_t^m, \cdots, \mathbf{y}_{T_m}^m]$, $\mathbf{Y}^m \in \mathcal{R}^{n \times T_m}$, $\mathbf{y}_t^m \in \mathcal{R}^{n \times 1}$.
- $\mathbf{Z}^m = [\mathbf{z}_1^m, \cdots, \mathbf{z}_t^m, \cdots, \mathbf{y}_{T_m}^m]$, $\mathbf{Z}^m \in \mathcal{R}^{d \times T_m}$, $\mathbf{z}_t^m \in \mathcal{R}^{d \times 1}$.
- n is the number of variables. d is the dimension of hidden state.
- T_m is the length of mth sequence.
- $\mathbf{Z}_+^m = [\mathbf{z}_2^m, \mathbf{z}_3^m, \cdots, \mathbf{z}_{T_m}^m]$ and $\mathbf{Z}_-^m = [\mathbf{z}_1^m, \mathbf{z}_2^m, \cdots, \mathbf{z}_{T_m-1}^m]$.
- We use $\mathbf{Y}$, $\mathbf{Z}$, $\mathbf{Z}_+$, and $\mathbf{Z}_-$ to denote the horizontal concatenations of $\{\mathbf{Y}^m\}$, $\{\mathbf{Z}^m\}$, $\{\mathbf{Z}_+^m\}$, and $\{\mathbf{Z}_-^m\}$.

Emission Matrix

$$\mathbf{y}_t = \boxed{C}\mathbf{z}_t + \zeta_t$$

$$\min_{C,\mathbf{Z}} \|\mathbf{Y} - C\mathbf{Z}\|_F^2$$

Transition Matrix

$$\mathbf{z}_t = \boxed{A}\mathbf{z}_{t-1} + \boldsymbol{\epsilon}_t$$

$$\min_{A,\mathbf{Z}} \|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2$$

gLDS Framework

Emission Matrix

Transition Matrix

$$\mathbf{y}_t = \boxed{C}\mathbf{z}_t + \zeta_t$$

$$\mathbf{z}_t = \boxed{A}\mathbf{z}_{t-1} + \boldsymbol{\epsilon}_t$$

$$\min_{C,\mathbf{Z}} \|\mathbf{Y} - C\mathbf{Z}\|_F^2$$

$$\min_{A,\mathbf{Z}} \|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2$$

$$\min_{A,C,\mathbf{Z}} \|\mathbf{Y} - C\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \alpha\mathcal{R}_C(C) + \beta\mathcal{R}_{\mathbf{Z}}(\mathbf{Z}) + \gamma\mathcal{R}_A(A)$$

Learning - Optimization of A, C, and **Z**

gLDS Framework:

$$\min_{A,C,\mathbf{Z}} \|\mathbf{Y} - C\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \alpha\mathcal{R}_C(C) + \beta\mathcal{R}_{\mathbf{Z}}(\mathbf{Z}) + \gamma\mathcal{R}_A(A)$$

- $\min_A \|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \gamma/\lambda\mathcal{R}_A(A)$
- $\min_C \|\mathbf{Y} - C\mathbf{Z}\|_F^2 + \alpha\mathcal{R}_C(C)$
- $\min_{\mathbf{Z}} \|\mathbf{Y} - C\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \beta\mathcal{R}_{\mathbf{Z}}(\mathbf{Z})$

gLDS Framework:

$$\min_{A,C,\mathbf{Z}} \|\mathbf{Y} - C\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \alpha\mathcal{R}_C(C) + \beta\mathcal{R}_{\mathbf{Z}}(\mathbf{Z}) + \gamma\mathcal{R}_A(A)$$

- $\hat{Q} = \frac{1}{T-N}(\hat{\mathbf{Z}}_+ - \hat{A}\hat{\mathbf{Z}}_-)(\hat{\mathbf{Z}}_+ - \hat{A}\hat{\mathbf{Z}}_-)^\top$
- $\hat{R} = \frac{1}{T}(\mathbf{Y} - \hat{C}\hat{\mathbf{Z}})(\mathbf{Y} - \hat{C}\hat{\mathbf{Z}})^\top$
- $\hat{\xi} = \frac{1}{N}\sum_{m=1}^{N} \hat{\mathbf{z}}_1^m$
- $\hat{\Psi} = \frac{1}{N}\sum_{m=1}^{N} \hat{\mathbf{z}}_1^m(\hat{\mathbf{z}}_1^m)^\top$

The Ridge Model (gLDS-ridge)

gLDS Framework:

$$\min_{A,C,\mathbf{Z}} \|\mathbf{Y} - C\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \alpha\mathcal{R}_C(C) + \beta\mathcal{R}_{\mathbf{Z}}(\mathbf{Z}) + \gamma\mathcal{R}_A(A)$$

Set $\mathcal{R}_C(C)$, $\mathcal{R}_A(A)$, and $\mathcal{R}_{\mathbf{Z}}(\mathbf{Z})$ to the square of Frobenius norm.

gLDS-ridge:

$$\min_{A,C,\mathbf{Z}} \|\mathbf{Y} - C\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \alpha||C||_F^2 + \beta\|\mathbf{Z}\|_F^2 + \gamma\|A\|_F^2$$

Existing Models in gLDS Framework

Existing models become special cases in gLDS framework:

- Regularized LDS [Liu and Hauskrecht, 2015]: a low-rank transition matrix.
- Stable LDS [Boots et al., 2007]: the largest singular value of transition matrix is no greater than 1.

Learning Regularized LDS (gLDS-low-rank)

By setting $\mathcal{R}_A(A) = \|A\|_F^2 + \frac{\lambda}{\gamma}\gamma_A\|A\|_*$, we have

$$\min_A \|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \gamma/\lambda\|A\|_F^2 + \gamma_A\|A\|_*$$

Easily be optimized by proximal gradient descent algorithm.

Learning Stable LDS (gLDS-stable)

By setting $\mathcal{R}_A(A) = \emptyset$, we have

$$\min_A \|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 \Leftrightarrow \min_a a^\top B a - 2q^\top a$$

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

Standard quadratic program! We can apply the same constraints generation techniques described in [Boots et al., 2007] to guarantee the stability.

The Smooth Model (gLDS-smooth)

We propose a temporal smoothing regularization, which penalizes the difference of predictive results, to achieve smooth forecasts.

Temporal smoothing regularization:

$$\mathcal{R}_{\mathcal{T}}^{m} = \frac{1}{2}\sum_{i=1}^{T_m}\sum_{j=1}^{T_m} w_{ij}^{m}\|\hat{\mathbf{y}}_i^m - \hat{\mathbf{y}}_j^m\|_2^2 = Tr[C\mathbf{Z}^m L^m (\mathbf{Z}^m)^\top C^\top]$$

$$\mathcal{R}_{\mathcal{T}} = \sum_{m=1}^{N} \mathcal{R}_{\mathcal{T}}^{m} = Tr[C\mathbf{Z}P\mathbf{Z}^\top C^\top]$$

The Smooth Model (gLDS-smooth)

gLDS-smooth = gLDS-ridge + Temporal smoothing regularization:

$$\min_{A,C,\mathbf{Z}} \|\mathbf{Y} - C\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \alpha\|C\|_F^2 + \beta\|\mathbf{Z}\|_F^2 + \gamma\|A\|_F^2 + \delta Tr[C\mathbf{Z}P\mathbf{Z}^\top C^\top]$$

Easily be optimized by coordinate gradient descent algorithm.

Data Sets

- Flour price data (*flourprice*). It is a monthly flour price indices data, which contains the flour price series in Buffalo, Minneapolis and Kansas City, from August 1972 to November 1980.

- Evap data (*evap*). The evaporation data contains the daily amounts of water evaporated, temperature, and barometric pressure from 10/11/1692 to 09/11/1693.

- H2O evap data (*h2o_evap*). It contains six MTS variables: the amount of evaporation, total global radiation, estimated net radiation, saturation deficit at max temperature, mean daily wind speed and saturation deficit at mean temperature.

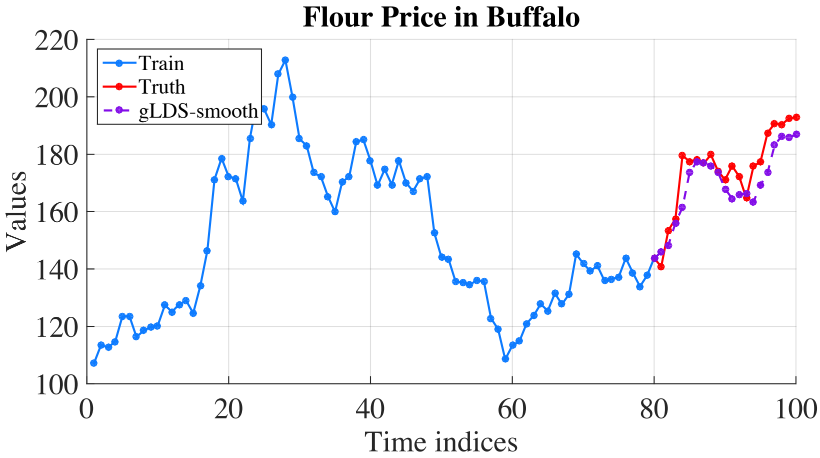
- Clinical data (*clinical*). A MTS clinical data obtained from electronic health records of post-surgical cardiac patients in PCP database.

Evaluation Metric - Mean Absolute Percentage Error

$$\text{MAPE} = \frac{|y_t - \hat{y}_t|}{y_t} \times 100\%$$

where $|\cdot|$ denotes the absolute value; y_t and $\hat{y}_t$ are the tth true and predicted values.

Qualitative Prediction Analysis



Quantitative Prediction Analysis

	Training: 80%		**Training: 90%**	
# of states	5	10	5	10
Spectral	24.62	24.85	25.08	26.28
EM	17.68	14.45	16.32	17.35
gLDS-ridge	10.58	10.35	13.60	14.05
gLDS-smooth	**10.35**	**10.27**	**13.39**	**13.68**

Table 1: Average-MAPE on *evap* dataset.

Stability Effects of gLDS-stable

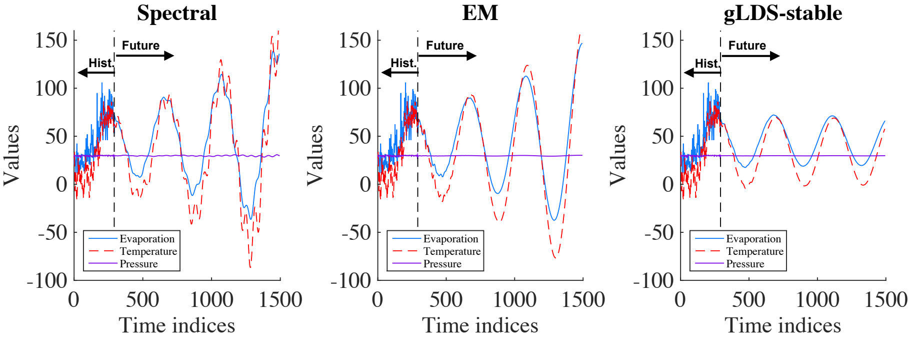


Figure 1: Training data and simulated sequences from gLDS-stable in *evap*.

Sparsification Effects of gLDS-low-rank

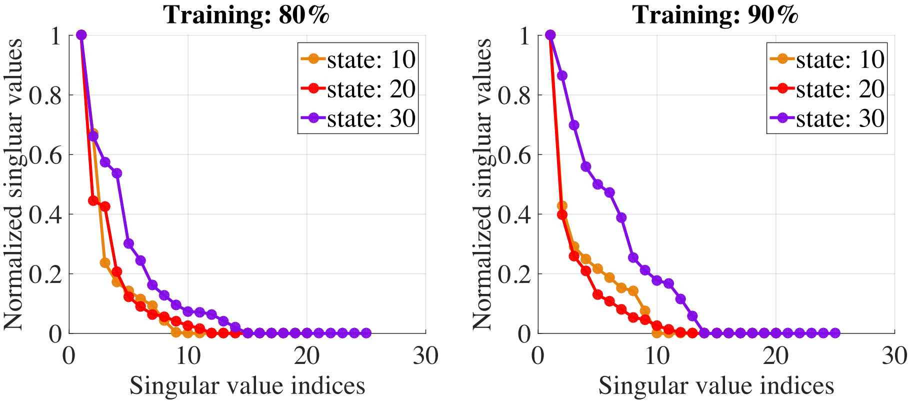


Figure 2: Intrinsic dimensionality recovery in *evap* dataset.

Conclusion

Advantages of our gLDS framework:

- a new approach to learn LDS from multiple MTS sequences
- easily incorporating constraints on both the hidden states and the parameters
- supporting accurate MTS prediction

Reference I

Boots, B., Gordon, G., and Siddiqi, S. (2007).
A constraint generation approach to learning stable linear dynamical systems.
In *NIPS*, pages 1329–1336.

Kalman, R. E. (1960).
A new approach to linear filtering and prediction problems.
Journal of Fluids Engineering, 82(1):35–45.

Liu, Z. and Hauskrecht, M. (2015).
A regularized linear dynamical system framework for multivariate time series analysis.
In *The 29th AAAI Conference on Artificial Intelligence*.

Thank you!

Q & A