Google

Universal Sequence Preconditioning

Annie Marsden

Elad Hazan

Aug 2022

General Overview

Aug 2022

Introduction

Which Problem Is Easier?

Observe: $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{t-1}$

Next Token Prediction	Next Token *Differences* Prediction
Predict: $\mathbf{y}_t$	Predict: $\mathbf{y}_t - \mathbf{y}_{t-1}$

Introduction

Which Problem Is Easier?

Observe: 1, 2, 3, 4, 5, 6

1, 1, 1, 1, 1, 1

Next Token Prediction

Next Token *Differences* Prediction

Predict: 7

Predict: 1

Introduction

Which Problem Is Easier?

Observe: 1, 2, 3, 4, 5, 6

Next Token Prediction

Predict: 7

1, 1, 1, 1, 1, 1

Next Token *Differences* Prediction

Predict: 1

Introduction

Which Problem Is Easier?

Observe: 1, 2, 3, 4, 5, 6

Next Token Prediction

Predict: 7

1, 1, 1, 1, 1, 1

Next Token *Differences* Prediction

Predict: 1

"Method of Differencing"
Box and Jenkins 1970s

Introduction

Which Problem Is Easier?

Observe: $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{t-1}$

Next Token Prediction	Next Token *Differences* Prediction	General Moving Average
$\mathbf{y}_t$	$\mathbf{y}_t - \mathbf{y}_{t-1}$	$\boxed{\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i}}$

Introduction

Q: Is this provable?

General theory for sequence preconditioning?

Introduction

Q: Is this provable?

General theory for sequence preconditioning?

A: Of course not...

$y_t = N(0,1)$

Introduction

Q: Is there a family of signals s.t. sequence preconditioning provably works?

Introduction

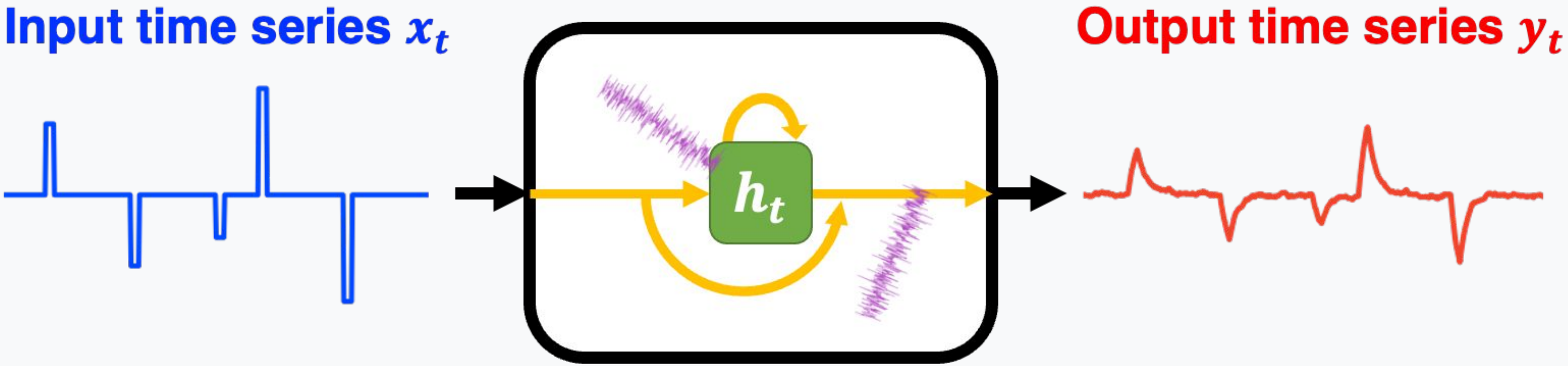
Q: Is there a family of signals s.t. sequence preconditioning provably works?

A: Linear Dynamical Systems

Introduction

Linear Dynamical Systems

Today we are talking about foundational research.



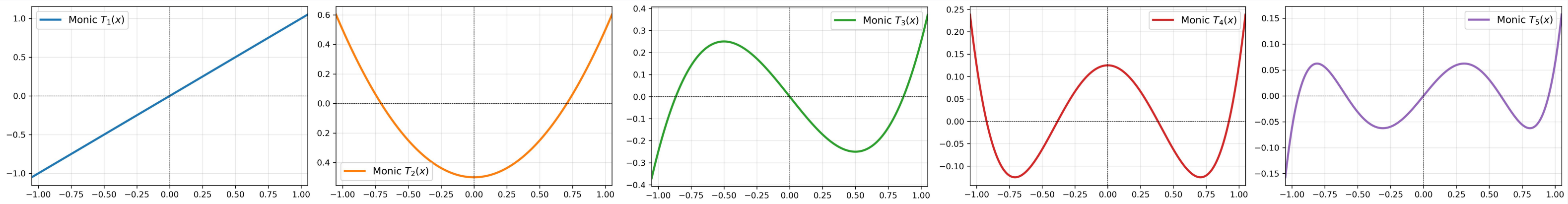
Sequence Preconditioning Guarantee

Informal Theorem (Marsden, Hazan)

Preconditioning the outputs with $c_1, \ldots, c_n$ chosen to be the coefficients of the n-th monic **Chebyshev polynomial** provably:

1) Improves the regret of several well-known learning algorithms.

2) Extends the class of linear dynamical systems these algorithms can learn.



Preconditioning Algorithm

Step One: Preconditioning

$$(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L) \xrightarrow{\text{Convolution}} (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_L)$$

Target for prediction

Preconditioned Target

$$\mathbf{z}_t \leftarrow \mathrm{conv}(\mathbf{y}_{t-n:t}, c_{1:n})$$

Preconditioning Algorithm

Step One: Preconditioning

$(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L) \xrightarrow{\text{Convolution}} (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_L)$

Target for prediction

Preconditioned Target

$$\mathbf{z}_t \leftarrow \mathrm{conv}(\mathbf{y}_{t-n:t}, c_{1:n})$$

Step Two: Training

Train your favorite algorithm on $(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_L)$

Get preconditioned predictions $(\hat{\mathbf{z}}_1, \hat{\mathbf{z}}_2, \ldots, \hat{\mathbf{z}}_L)$

Preconditioning Algorithm

Step One: Preconditioning

$(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L) \xrightarrow{\text{Convolution}} (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_L)$

Target for prediction

Preconditioned Target

$$\mathbf{z}_t \leftarrow \mathrm{conv}(\mathbf{y}_{t-n:t}, c_{1:n})$$

Step Two: Training

Train your favorite algorithm on $(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_L)$

Get preconditioned predictions $(\hat{\mathbf{z}}_1, \hat{\mathbf{z}}_2, \ldots, \hat{\mathbf{z}}_L)$

Step Three: Convert Predictions to Original Signal

$(\hat{\mathbf{z}}_1, \hat{\mathbf{z}}_2, \ldots, \hat{\mathbf{z}}_L) \xrightarrow{\textbf{Deconvolution}} (\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_L)$

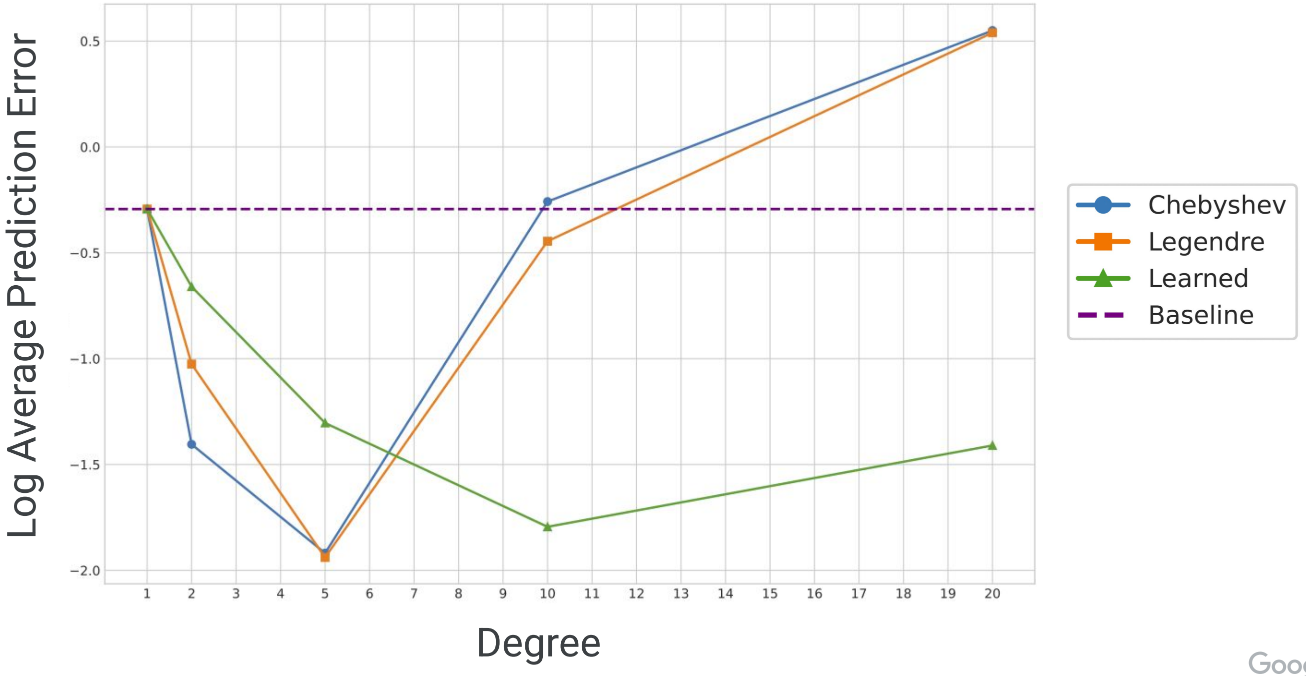
$$\hat{\mathbf{y}}_t \leftarrow \mathrm{deconv}(\mathbf{z}_{1:t}, c_{1:n})$$

Experiments

Synthetic Datasets

Algorithm: Regression

Dataset: LDS

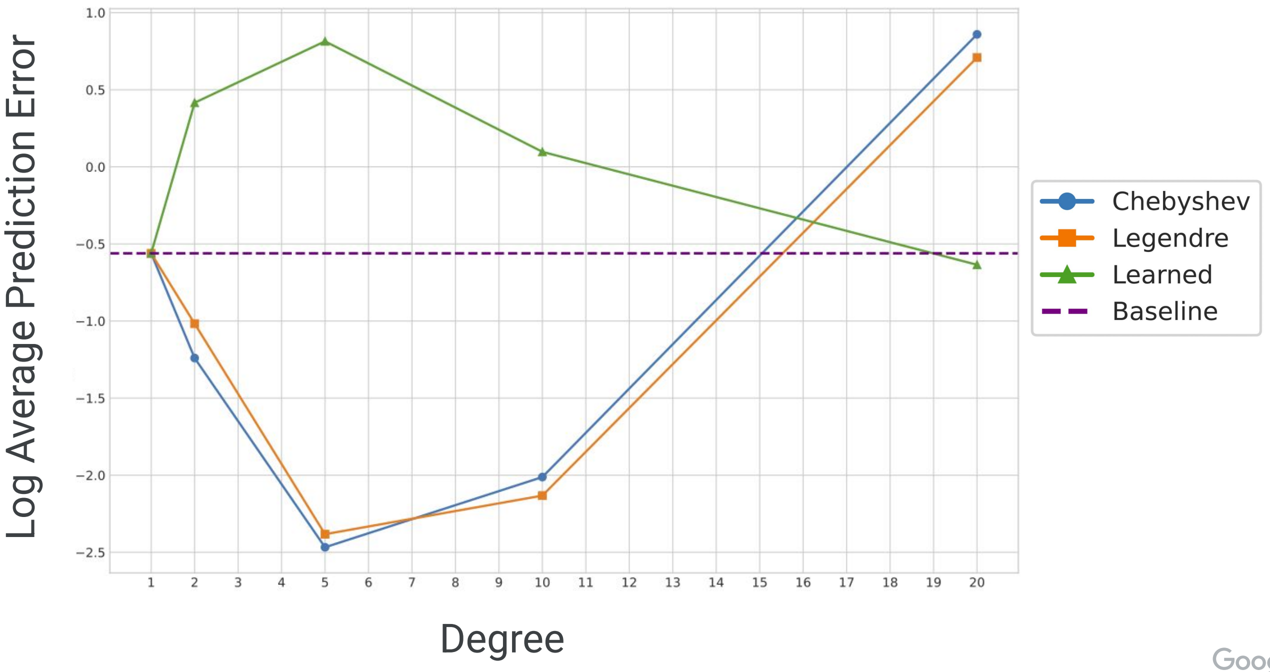


Experiments

Synthetic Datasets

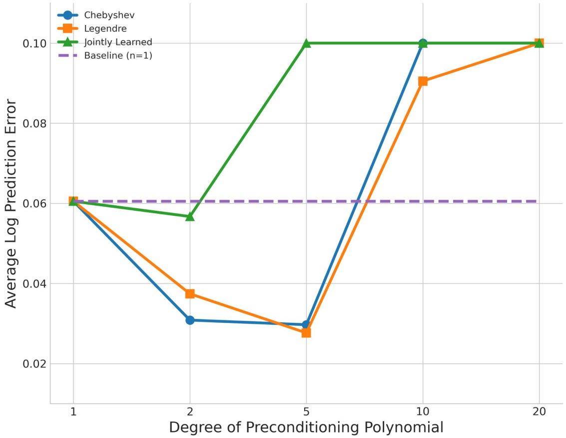
Algorithm: 10-layer DNN

Dataset: 10-layer DNN

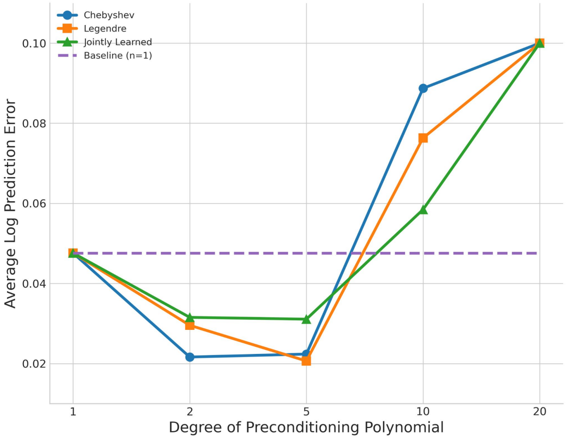


Experiments

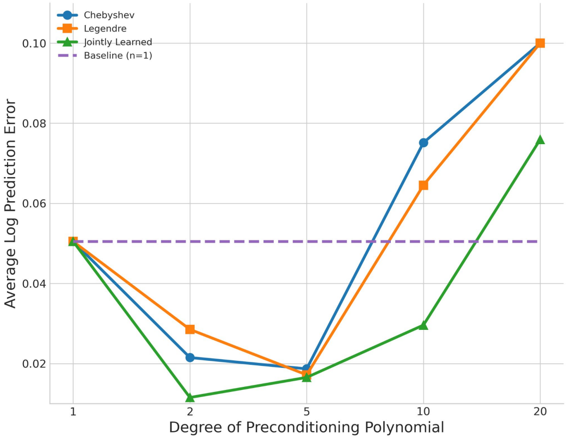
ETTh1 Dataset



T=1000



T=2500



T=5000

Theoretical Understanding

Aug 2022

Online Sequential Prediction Setting

$\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{t-1} \longleftarrow$ history of inputs

$\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{t-1} \longleftarrow$ history of outputs

Observe: $\mathbf{u}_t$

Predict: $\hat{\mathbf{y}}_t$

Observe true $\mathbf{y}_t$ and suffer loss

Understanding Universal Sequence Preconditioning

Linear Dynamical System

$$\mathbf{h}_t = \mathbf{A}\mathbf{h}_{t-1} + \mathbf{B}\mathbf{u}_t \qquad \mathbf{y}_t = \mathbf{C}\mathbf{h}_t + \mathbf{D}\mathbf{u}_t$$

hidden state

hidden transition matrix

WLOG can ignore

Understanding Universal Sequence Preconditioning

Linear Dynamical System

$$\mathbf{h}_t = \mathbf{A}\mathbf{h}_{t-1} + \mathbf{B}\mathbf{u}_t \qquad \mathbf{y}_t = \mathbf{C}\mathbf{h}_t + \mathbf{D}\mathbf{u}_t$$

hidden state

hidden transition matrix

WLOG can ignore

Dominant notions of complexity

Horizon T

Hidden dimension d

Allowing **A** to be asymmetric

Understanding Universal Sequence Preconditioning

Linear Dynamical System

$$\mathbf{h}_t = \mathbf{A}\mathbf{h}_{t-1} + \mathbf{B}\mathbf{u}_t \qquad \mathbf{y}_t = \mathbf{C}\mathbf{h}_t + \mathbf{D}\mathbf{u}_t$$

hidden state

hidden transition matrix

WLOG can ignore

Factor out the hidden state:

$$\mathbf{y}_t = \sum_{s=0}^{t} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-s}$$

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t = \sum_{s=0}^{t} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-s}$$

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t = \sum_{s=0}^{t} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-s}$$

$$\mathbf{y}_t + c_1\mathbf{y}_{t-1} + c_2\mathbf{y}_{t-2} = \left(\sum_{s=0}^{t} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-s}\right) + c_1\left(\sum_{s=0}^{t-1} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-1-s}\right) + c_2\left(\sum_{s=0}^{t-2} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-2-s}\right)$$

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t = \sum_{s=0}^{t} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-s}$$

$$\mathbf{y}_t + c_1\mathbf{y}_{t-1} + c_2\mathbf{y}_{t-2} = \left(\sum_{s=0}^{t} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-s}\right) + c_1\left(\sum_{s=0}^{t-1} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-1-s}\right) + c_2\left(\sum_{s=0}^{t-2} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-2-s}\right)$$

$$= \mathbf{C}\mathbf{B}\mathbf{u}_t + (\mathbf{C}\mathbf{A}\mathbf{B} + c_1\mathbf{C}\mathbf{B})\mathbf{u}_{t-1} + \left(\mathbf{C}\mathbf{A}^2\mathbf{B} + c_1\mathbf{C}\mathbf{A}\mathbf{B} + c_2\mathbf{B}\mathbf{C}\right)\mathbf{u}_{t-2} + \ldots$$

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t = \sum_{s=0}^{t} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-s}$$

$$\mathbf{y}_t + c_1\mathbf{y}_{t-1} + c_2\mathbf{y}_{t-2} = \left(\sum_{s=0}^{t} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-s}\right) + c_1\left(\sum_{s=0}^{t-1} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-1-s}\right) + c_2\left(\sum_{s=0}^{t-2} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-2-s}\right)$$

$$= \mathbf{CBu}_t + (\mathbf{CAB} + c_1\mathbf{CB})\mathbf{u}_{t-1} + \big(\underbrace{\mathbf{CA}^2\mathbf{B} + c_1\mathbf{CAB} + c_2\mathbf{BC}}_{}\big)\mathbf{u}_{t-2} + \ldots$$

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t = \sum_{s=0}^{t} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-s}$$

$$\mathbf{y}_t + c_1\mathbf{y}_{t-1} + c_2\mathbf{y}_{t-2} = \left(\sum_{s=0}^{t} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-s}\right) + c_1\left(\sum_{s=0}^{t-1} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-1-s}\right) + c_2\left(\sum_{s=0}^{t-2} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-2-s}\right)$$

$$= \mathbf{C}\mathbf{B}\mathbf{u}_t + (\mathbf{C}\mathbf{A}\mathbf{B} + c_1\mathbf{C}\mathbf{B})\mathbf{u}_{t-1} + \underbrace{\left(\mathbf{C}\mathbf{A}^2\mathbf{B} + c_1\mathbf{C}\mathbf{A}\mathbf{B} + c_2\mathbf{B}\mathbf{C}\right)}_{\mathbf{C}\left(\mathbf{A}^2 + c_1\mathbf{A} + c_2\right)\mathbf{B}}\mathbf{u}_{t-2} + \ldots$$

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t = \sum_{s=0}^{t} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-s}$$

$$\mathbf{y}_t + c_1\mathbf{y}_{t-1} + c_2\mathbf{y}_{t-2} = \left(\sum_{s=0}^{t} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-s}\right) + c_1\left(\sum_{s=0}^{t-1} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-1-s}\right) + c_2\left(\sum_{s=0}^{t-2} \mathbf{C}\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-2-s}\right)$$

$$= \mathbf{C}\mathbf{B}\mathbf{u}_t + (\mathbf{C}\mathbf{A}\mathbf{B} + c_1\mathbf{C}\mathbf{B})\mathbf{u}_{t-1} + \underbrace{\left(\mathbf{C}\mathbf{A}^2\mathbf{B} + c_1\mathbf{C}\mathbf{A}\mathbf{B} + c_2\mathbf{B}\mathbf{C}\right)}_{\mathbf{C}\,\underbrace{\left(\mathbf{A}^2 + c_1\mathbf{A} + c_2\right)}_{p_n^c(\mathbf{A})}\mathbf{B}}\mathbf{u}_{t-2} + \ldots$$

$$p_n^c(x) := x^2 + c_1 x + c_2$$

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t = \sum_{s=0}^{t} \mathbf{CA}^s\mathbf{Bu}_{t-s}$$

$$\mathbf{y}_t + c_1\mathbf{y}_{t-1} + c_2\mathbf{y}_{t-2} = \left(\sum_{s=0}^{t} \mathbf{CA}^s\mathbf{Bu}_{t-s}\right) + c_1\left(\sum_{s=0}^{t-1} \mathbf{CA}^s\mathbf{Bu}_{t-1-s}\right) + c_2\left(\sum_{s=0}^{t-2} \mathbf{CA}^s\mathbf{Bu}_{t-2-s}\right)$$

$$= \mathbf{CBu}_t + (\mathbf{CAB} + c_1\mathbf{CB})\mathbf{u}_{t-1} + \underbrace{(\mathbf{CA}^2\mathbf{B} + c_1\mathbf{CAB} + c_2\mathbf{BC})}\mathbf{u}_{t-2} + \ldots$$

$$\mathbf{C}\underbrace{\left(\mathbf{A}^2 + c_1\mathbf{A} + c_2\right)}\mathbf{B}$$

$$p_n^c(x) := x^2 + c_1 x + c_2 \qquad p_n^c(\mathbf{A})$$

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t = \sum_{s=0}^{t} \mathbf{CA}^s\mathbf{Bu}_{t-s}$$

$$\mathbf{y}_t + c_1\mathbf{y}_{t-1} + c_2\mathbf{y}_{t-2} = \left(\sum_{s=0}^{t} \mathbf{CA}^s\mathbf{Bu}_{t-s}\right) + c_1\left(\sum_{s=0}^{t-1} \mathbf{CA}^s\mathbf{Bu}_{t-1-s}\right) + c_2\left(\sum_{s=0}^{t-2} \mathbf{CA}^s\mathbf{Bu}_{t-2-s}\right)$$

$$= \mathbf{CBu}_t + (\mathbf{CAB} + c_1\mathbf{CB})\mathbf{u}_{t-1} + \underbrace{(\mathbf{CA}^2\mathbf{B} + c_1\mathbf{CAB} + c_2\mathbf{BC})}_{\mathbf{C}\underbrace{\left(\mathbf{A}^2 + c_1\mathbf{A} + c_2\right)}_{p_n^c(\mathbf{A})}\mathbf{B}}\mathbf{u}_{t-2} + \dots$$

$$p_n^c(x) := x^2 + c_1x + c_2$$

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i} = \sum_{s=0}^{n-1} \left(\sum_{i=0}^{s} c_i \mathbf{C}\mathbf{A}^{s-i}\mathbf{B} \right) \mathbf{u}_{t-s} + \sum_{s=0}^{t-n} \mathbf{C} p_n^c(\mathbf{A}) \mathbf{A}^s \mathbf{B} \mathbf{u}_{t-n-s}$$

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i} = \underbrace{\sum_{s=0}^{n-1} \left(\sum_{i=0}^{s} c_i \mathbf{C} \mathbf{A}^{s-i} \mathbf{B} \right) \mathbf{u}_{t-s}}_{\text{n most recent observations}} + \sum_{s=0}^{t-n} \mathbf{C} p_n^c(\mathbf{A}) \mathbf{A}^s \mathbf{B} \mathbf{u}_{t-n-s}$$

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i} = \underbrace{\sum_{s=0}^{n-1} \left(\sum_{i=0}^{s} c_i \mathbf{C} \mathbf{A}^{s-i} \mathbf{B} \right) \mathbf{u}_{t-s}}_{\text{n most recent observations}} + \underbrace{\sum_{s=0}^{t-n} \mathbf{C} p_n^c(\mathbf{A}) \mathbf{A}^s \mathbf{B} \mathbf{u}_{t-n-s}}_{\text{grows with t}}$$

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i} = \sum_{s=0}^{n-1} \left(\sum_{i=0}^{s} c_i \mathbf{C} \mathbf{A}^{s-i} \mathbf{B} \right) \mathbf{u}_{t-s} + \underbrace{\sum_{s=0}^{t-n} \mathbf{C} p_n^c(\mathbf{A}) \mathbf{A}^s \mathbf{B} \mathbf{u}_{t-n-s}}$$

Suppose this is the
characteristic polynomial of A

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i} = \sum_{s=0}^{n-1} \left(\sum_{i=0}^{s} c_i \mathbf{C}\mathbf{A}^{s-i}\mathbf{B} \right) \mathbf{u}_{t-s} + \sum_{s=\cdots}^{t-\cdots} \mathbf{C} \cdots \mathbf{B}\mathbf{u}_{t-n-s}$$

Cayley-Hamilton

Suppose this is the characteristic polynomial of A

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i} = \sum_{s=0}^{n-1} \left(\sum_{i=0}^{s} c_i \mathbf{C} \mathbf{A}^{s-i} \mathbf{B} \right) \mathbf{u}_{t-s}$$

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i} = \sum_{s=0}^{n-1} \left(\sum_{i=0}^{s} c_i \mathbf{C}\mathbf{A}^{s-i}\mathbf{B} \right) \mathbf{u}_{t-s}$$

Preconditioned signal

Could just learn this with regression.

$$\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i} = \sum_{s=0}^{n-1} \mathbf{M}_s \mathbf{u}_{t-s}$$

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i} = \sum_{s=0}^{n-1} \left(\sum_{i=0}^{s} c_i \mathbf{C} \mathbf{A}^{s-i} \mathbf{B} \right) \mathbf{u}_{t-s}$$

Problem One: Hidden transition matrix is unknown

Problem Two: Induces a dependence on the hidden dimension (which could be prohibitively large)

Preconditioned signal

Could just learn this with regression.

$$\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i} = \sum_{s=0}^{n-1} \mathbf{M}_s \mathbf{u}_{t-s}$$

Solution?

Solution?

Chebyshev polynomial

Why Chebyshev?

The n-th degree (monic) chebyshev polynomial **universally** shrinks the region [-1,1]:

$$\max_{x\in[-1,1]} \left|p_n^{\text{chebyshev}}(x)\right| = \frac{1}{2^{n-1}}$$

TLDR: Don't worry about learning A! This will work regardless of the true underlying system…

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i} = \sum_{s=0}^{n-1} \left(\sum_{i=0}^{s} c_i \mathbf{C}\mathbf{A}^{s-i}\mathbf{B} \right) \mathbf{u}_{t-s} + \sum_{s=0}^{t-n} \mathbf{C} p_n^c(\mathbf{A}) \mathbf{A}^s \mathbf{B} \mathbf{u}_{t-n-s}$$

$$T \cdot 2^{-n}$$

Almost as good as gone…

Convolution of target converts to a polynomial in the hidden space

$$\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i} = \sum_{s=0}^{n-1} \left(\sum_{i=0}^{s} c_i \mathbf{C}\mathbf{A}^{s-i}\mathbf{B} \right) \mathbf{u}_{t-s} + \sum_{s=0}^{t-n} \mathbf{C} p_n^c(\mathbf{A}) \mathbf{A}^s \mathbf{B} \mathbf{u}_{t-n-s}$$

$2^{0.3n}$

$T \cdot 2^{-n}$

Almost as good as gone...

Original USP Algorithm

OGD

$$\hat{y}_t(\theta) \leftarrow -\sum_{i=1}^{n} c_i y_{t-i} + \sum_{s=0}^{n-1} \theta_s \mathbf{u}_{t-s}$$

$$\theta^{(t+1)} \leftarrow \Pi_{\mathcal{K}} \left(\theta^{(t)} - \eta_t \nabla_\theta \ell(y_t, \hat{y}_t(\theta)) \right)$$

$$\mathcal{K} = \{(\theta_0, \ldots, \theta_{n-1}) \text{ s.t. } \|\theta_s\| \leq R\}$$

Original USP Algorithm

OGD

$$\hat{y}_t(\theta) \leftarrow -\sum_{i=1}^{n} c_i y_{t-i} + \sum_{s=0}^{n-1} \theta_s \mathbf{u}_{t-s}$$

$$\theta^{(t+1)} \leftarrow \Pi_{\mathcal{K}} \left(\theta^{(t)} - \eta_t \nabla_\theta \ell(y_t, \hat{y}_t(\theta))\right)$$

$$\mathcal{K} = \{(\theta_0, \ldots, \theta_{n-1}) \text{ s.t. } \|\theta_s\| \leq R\}$$

We want our domain to contain the true solution

$$\theta_s^* = \sum_{i=0}^{s} c_i \mathbf{C} \mathbf{A}^{s-i} \mathbf{B}$$

R must grow exponentially with n

Original USP Algorithm

OGD

$$\hat{y}_t(\theta) \leftarrow -\sum_{i=1}^{n} c_i y_{t-i} + \sum_{s=0}^{n-1} \theta_s \mathbf{u}_{t-s}$$

$$\theta^{(t+1)} \leftarrow \Pi_{\mathcal{K}}\left(\theta^{(t)} - \eta_t \nabla_\theta \ell(y_t, \hat{y}_t(\theta))\right)$$

$$\mathcal{K} = \{(\theta_0, \ldots, \theta_{n-1}) \text{ s.t. } \|\theta_s\| \leq R\}$$

OGD Regret

$$GR/\sqrt{T}$$

We want our domain to contain the true solution

$$\theta_s^* = \sum_{i=0}^{s} c_i \mathbf{C}\mathbf{A}^{s-i}\mathbf{B}$$

R must grow exponentially with n

Original USP Algorithm

OGD

$$\hat{y}_t(\theta) \leftarrow -\sum_{i=1}^{n} c_i y_{t-i} + \sum_{s=0}^{n-1} \theta_s \mathbf{u}_{t-s}$$

$$\theta^{(t+1)} \leftarrow \Pi_{\mathcal{K}} \left(\theta^{(t)} - \eta_t \nabla_\theta \ell(y_t, \hat{y}_t(\theta)) \right)$$

$$\mathcal{K} = \{(\theta_0, \ldots, \theta_{n-1}) \text{ s.t. } \|\theta_s\| \le R\}$$

OGD Regret

$$2^{0.6n} / \sqrt{T}$$

We want our domain to contain the true solution

$$\theta_s^* = \sum_{i=0}^{s} c_i \mathbf{C} \mathbf{A}^{s-i} \mathbf{B}$$

R must grow exponentially with n

Original USP Result

Regret using OGD:

$$\frac{1}{T}\sum_{t=1}^{T}\|\hat{\mathbf{y}}_t - \mathbf{y}_t\|_1 \;\leq\; \tilde{O}\left(\frac{\|\mathbf{B}\|\|\mathbf{C}\|\kappa\sqrt{d_{out}}}{T^{2/13}}\right)$$

No hidden dimension
Allows A to be asymmetric

Feels like it misses the full power…

USP Summary

$$\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i} = \sum_{s=0}^{n-1} \left(\sum_{i=0}^{s} c_i \mathbf{C} \mathbf{A}^{s-i} \mathbf{B} \right) \mathbf{u}_{t-s} + \sum_{s=0}^{t-n} \mathbf{C} p_n^c(\mathbf{A}) \mathbf{A}^s \mathbf{B} \mathbf{u}_{t-n-s}$$

$$y_t = -\sum_{i=1}^{n} c_i y_{t-i} + \sum_{s=0}^{n-1} \theta_s^{\mathrm{USP}} u_{t-s} + \epsilon_t$$

Existence of a good solution

USP Summary

$$\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i} = \sum_{s=0}^{n-1} \left(\sum_{i=0}^{s} c_i \mathbf{C}\mathbf{A}^{s-i}\mathbf{B} \right) \mathbf{u}_{t-s} + \sum_{s=0}^{t-n} \mathbf{C} p_n^c(\mathbf{A}) \mathbf{A}^s \mathbf{B} \mathbf{u}_{t-n-s}$$

Huge Norm

$$y_t = -\sum_{i=1}^{n} c_i y_{t-i} + \sum_{s=0}^{n-1} \theta_s^{\text{USP}} u_{t-s} + \epsilon_t$$

Highly Compressed Dimension

USP Summary

$$\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i} = \sum_{s=0}^{n-1}\left(\sum_{i=0}^{s} c_i \mathbf{C}\mathbf{A}^{s-i}\mathbf{B}\right)\mathbf{u}_{t-s} + \sum_{s=0}^{t-n} \mathbf{C}p_n^c(\mathbf{A})\mathbf{A}^s\mathbf{B}\mathbf{u}_{t-n-s}$$

Huge Norm

$$y_t = -\sum_{i=1}^{n} c_i y_{t-i} + \sum_{s=0}^{n-1} \theta_s^{\mathrm{USP}} u_{t-s} + \epsilon_t$$

Highly Compressed Dimension

First Order learning algorithms treat these things as the same. R grows linearly with both.

USP Summary

$$\mathbf{y}_t + \sum_{i=1}^{n} c_i \mathbf{y}_{t-i} = \sum_{s=0}^{n-1} \left(\sum_{i=0}^{s} c_i \mathbf{C}\mathbf{A}^{s-i}\mathbf{B} \right) \mathbf{u}_{t-s} + \sum_{s=0}^{t-n} \mathbf{C} p_n^c(\mathbf{A}) \mathbf{A}^s \mathbf{B} \mathbf{u}_{t-n-s}$$

Huge Norm

Second Order– Dimension and Radius are *very* different!

$$y_t = -\sum_{i=1}^{n} c_i y_{t-i} + \sum_{s=0}^{n-1} \theta_s^{\mathrm{USP}} u_{t-s} + \epsilon_t$$

Highly Compressed Dimension

First Order learning algorithms treat these things as the same. R grows linearly with both.

Vovk-Azoury-Warmuth Algorithm

Theorem 1 (VAW Regret). *Assume the true labels are uniformly bounded by Y, i.e. $|y_t| \leq Y$ for all $t = 1, \ldots, T$. Assume the feature vectors are uniformly bounded by R, i.e. $||\boldsymbol{a}_t||_2 \leq R$ for all $t = 1, \ldots T$. For any benchmark $\boldsymbol{x}^* \in \mathbb{R}^n$, the regret is bounded by:*

$$\sum_{t=1}^{T} \frac{1}{2}(\hat{y}_t - y_t)^2 \leq \underbrace{\sum_{t=1}^{T} \frac{1}{2}({\boldsymbol{x}^*}^\top \boldsymbol{a}_t - y_t)^2}_{\text{Prediction Error: } T \cdot 2^{-n}} + \frac{1}{2} + \frac{Y^2}{2} n \log\left(1 + \overset{2^{0.3n}}{\frac{T\|\boldsymbol{x}^*\|_2^2 R^2}{n}}\right) \tag{5}$$

Linear dependence on compressed dimension

Logarithmic dependence on norm!

USP Final Result

Theorem 2 (Main Result)**.** *Suppose the signal $y_1, \ldots, y_T$ comes from a linear dynamical system parameterized by $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ as in Eq. 1 where $\mathbf{A}$ has eigenvalues contained in the set $\left\{z \in \mathbb{C} \text{ s.t. } |z| \leq 1 \text{ and } |\arg(z)| \leq (1/24^2)\right\}$ and is diagonalized by matrix $\boldsymbol{P}$. Let $\kappa = ||\boldsymbol{P}||||\boldsymbol{P}^{-1}||$. Assume the signal $y_1, \ldots, y_T$ is bounded by Y, i.e. $\max_{t\in[T]}|y_t| \leq Y$. Further assume the inputs are bounded by 1, i.e. $\max_{t\in[T]}|u_t| \leq 1$. If $n = 3\log(||\boldsymbol{C}|| \cdot ||\boldsymbol{B}|| \cdot \kappa \cdot T)$ then the predictions $\hat{y}_1, \ldots, \hat{y}_T$ from Algorithm 2 have cumulative prediction loss,*

$$\sum_{t=1}^{T} (\hat{y}_t - y_t)^2 \leq O\left(Y^2 \log^3(1 + TY\|\boldsymbol{C}\|_2\|\boldsymbol{B}\|_2\kappa)\right). \tag{8}$$

USP Final Result

Theorem 2 (Main Result) … *linear dynamical sys-*
tem parameterized by ($\boldsymbol{A}$ … *contained in the set*
$\{z \in \mathbb{C}$ *s.t.* $|z| \leq 1$ *and* $|\mathrm{ar}$ … $\boldsymbol{P}$. *Let* $\kappa = ||\boldsymbol{P}|| ||\boldsymbol{P}^{-1}||$.
Assume the signal $y_1, \ldots, y$ … *r assume the inputs are*
bounded by 1, *i.e.* $\max_{t \in [T]}$ … *predictions* $\hat{y}_1, \ldots, \hat{y}_T$
from Algorithm 2 *have cum* …

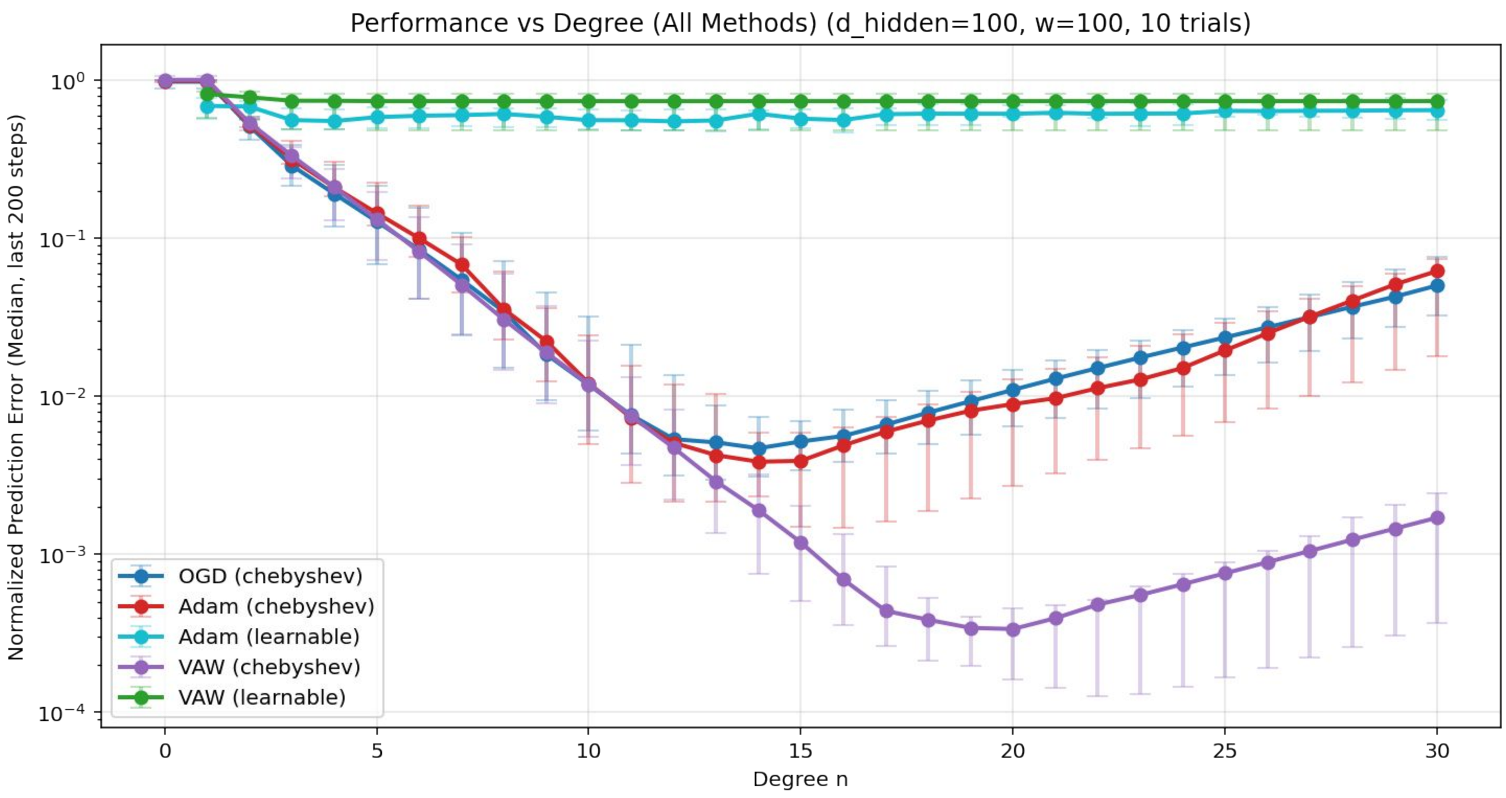
$$\sum_{t=1}^{T} (\ldots \tag{8}$$

<u>TLDR</u>

A can be asymmetric

No hidden dimension dependence

Poly-logarithmic Regret



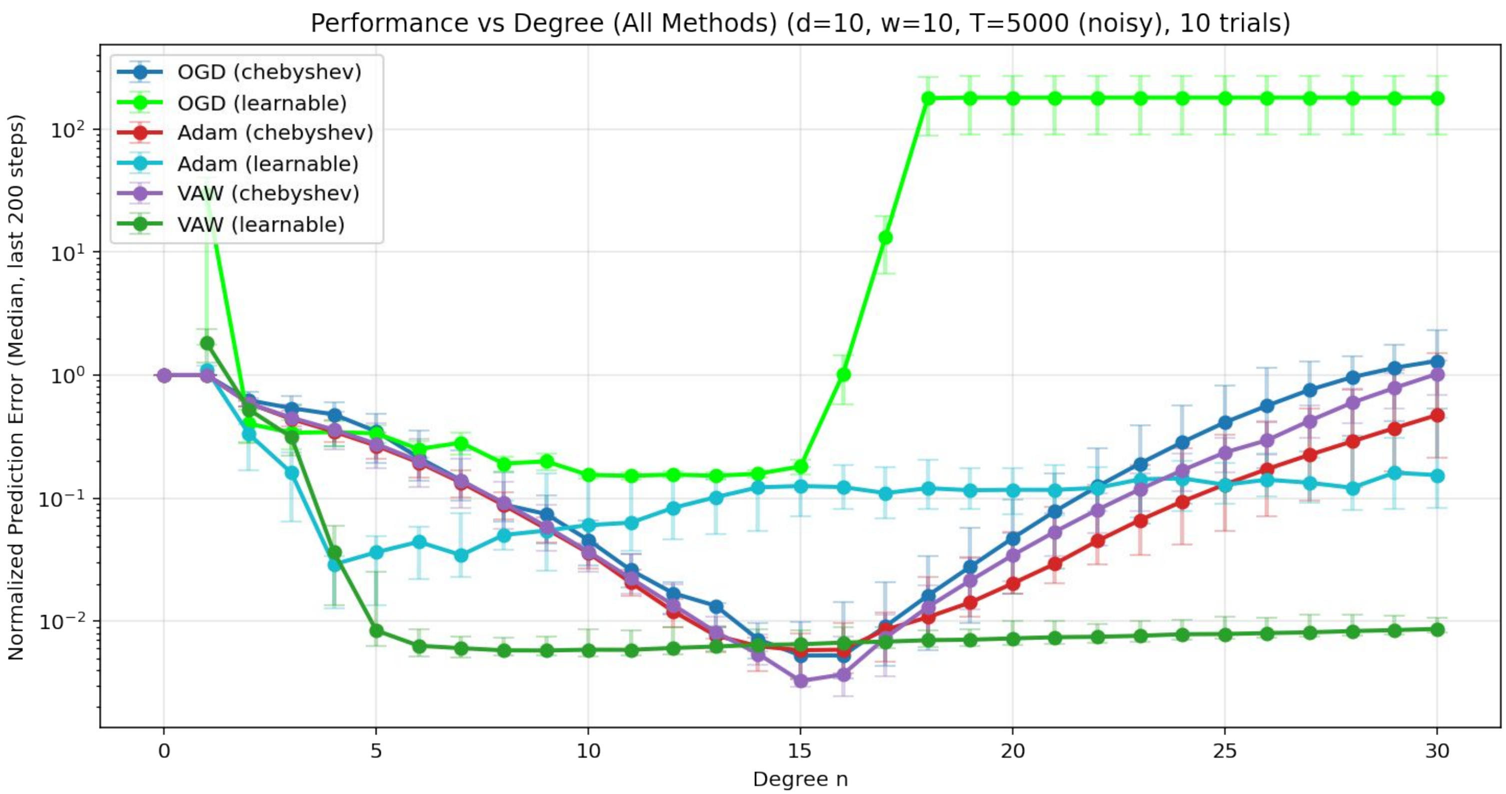
Performance vs Degree (All Methods) (d_hidden=100, w=100, 10 trials)
Normalized Prediction Error (Median, last 200 steps)
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
0
5
10
15
20
25
30
Degree n
OGD (chebyshev)
Adam (chebyshev)
Adam (learnable)
VAW (chebyshev)
VAW (learnable)

Future Work

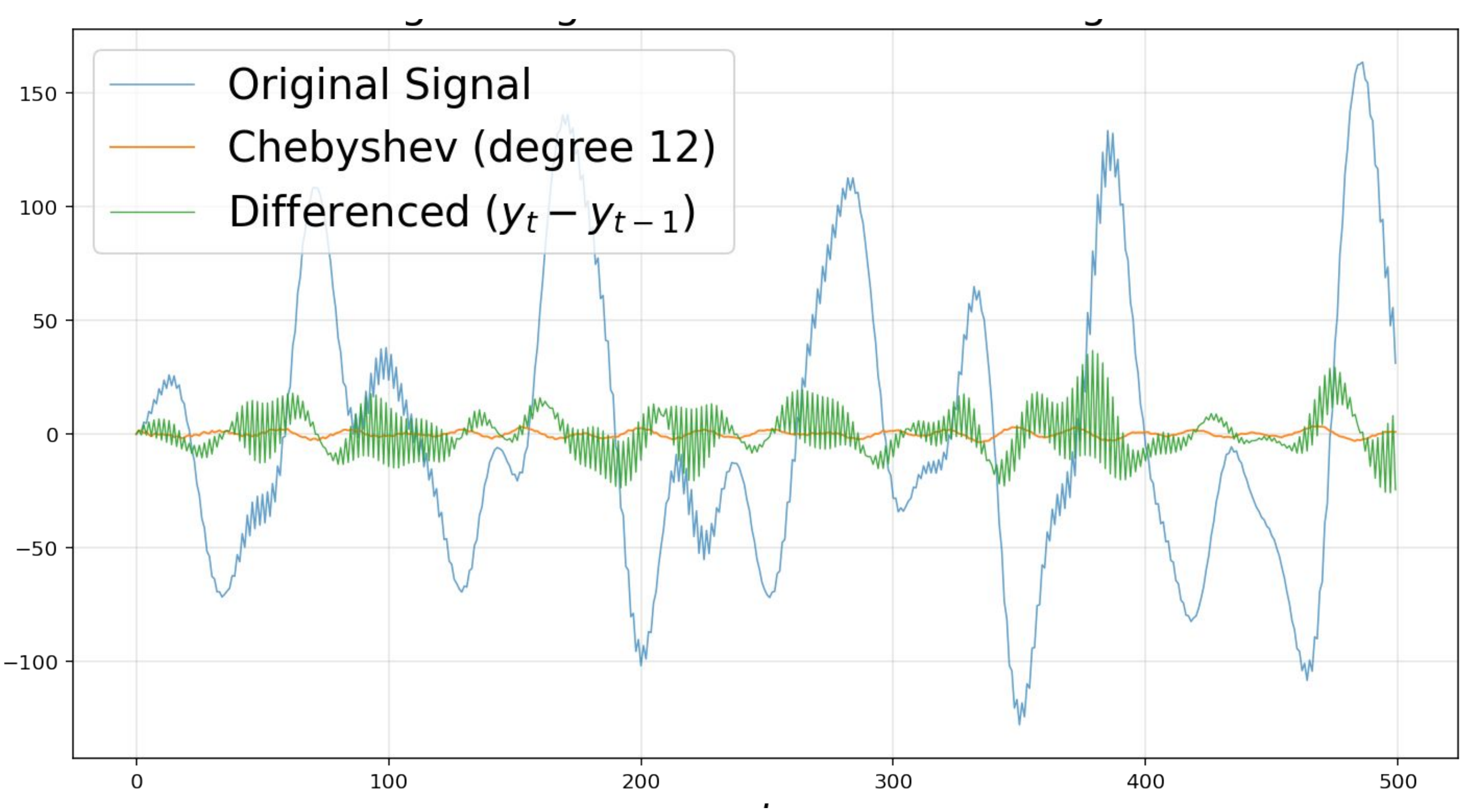
Practical: A lot of applications of this have shown great results.

Theoretical: Tight? Not with respect to the spectrum of **A.** For that we need to analyze the Faber polynomial.

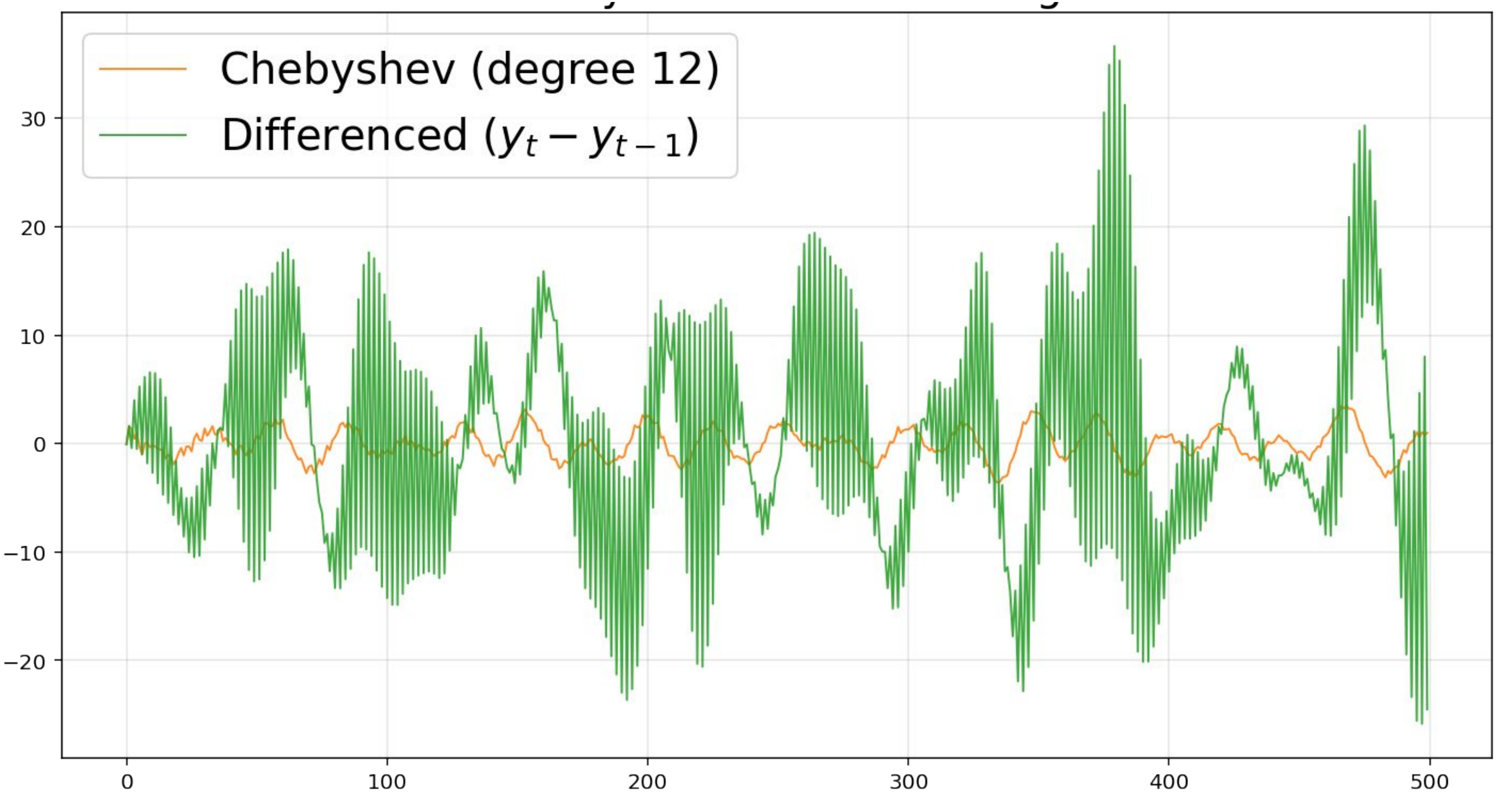
Questions?


Performance vs Degree (All Methods) (d=10, w=10, T=5000 (noisy), 10 trials)
OGD (chebyshev)
OGD (learnable)
Adam (chebyshev)
Adam (learnable)
VAW (chebyshev)
VAW (learnable)
Normalized Prediction Error (Median, last 200 steps)
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
0
5
10
15
20
25
30
Degree n

Google



Original Signal
Chebyshev (degree 12)
Differenced ($y_t - y_{t-1}$)
150
100
50
0
−50
−100
0
100
200
300
400
500


Chebyshev (degree 12)
Differenced ($y_t - y_{t-1}$)
30
20
10
0
−10
−20
0
100
200
300
400
500