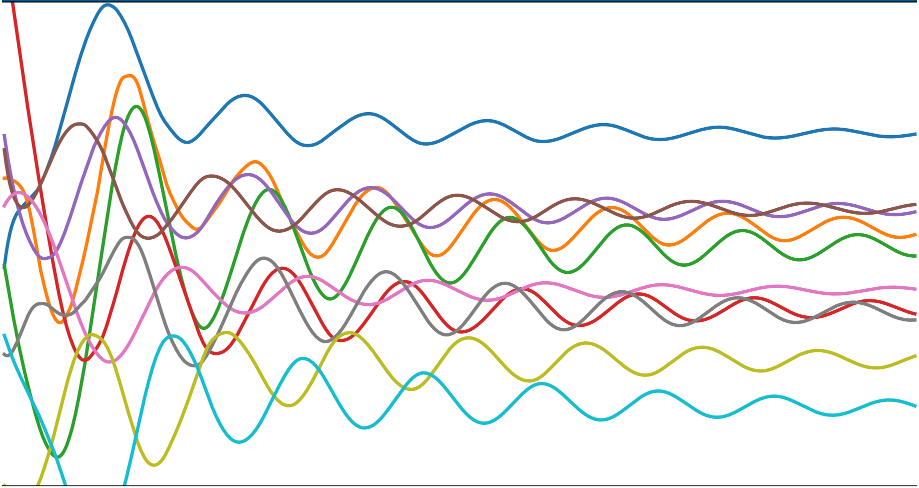
Variational Autoenconders for Koopman Dynamical Systems

B.Sc. Intermediate Presentation



Motivation

- Control theory for linear systems is highly evolved.
- But nonlinear systems are hard…
- We need ways to linearize a nonlinear system!

Motivation

- Control theory for linear systems is highly evolved.
- But nonlinear systems are hard…
- We need ways to linearize a nonlinear system!
- “Classical” Linearization:
  - Use small angle approximation (e.g. for pendulum)?
  - Diverge fast with higher displacements…
  - Only linearize locally, not globally.

Motivation

- Control theory for linear systems is highly evolved.
- But nonlinear systems are hard…
- We need ways to linearize a nonlinear system!
- “Classical” Linearization:
  - Use small angle approximation (e.g. for pendulum)?
  - Diverge fast with higher displacements…
  - Only linearize locally, not globally.
- Koopman theory offers a way to do that!

The Koopman Operator

For a *nonlinear* dynamical system

$$\mathsf{s}_{t+1} = \mathsf{F}(\mathsf{s}_t)$$

with *nonlinear* measurements (and *embedding*)

$$\mathsf{y}_t = \mathsf{g}(\mathsf{s}_t)$$

the Koopman operator advances these measurements forward in time *linearly*:

$$\mathsf{g}(\mathsf{s}_{t+1}) = \mathcal{K}\mathsf{g}(\mathsf{s}_t) \quad \Longleftrightarrow \quad \mathsf{y}_{t+1} = \mathcal{K}\mathsf{y}_t$$

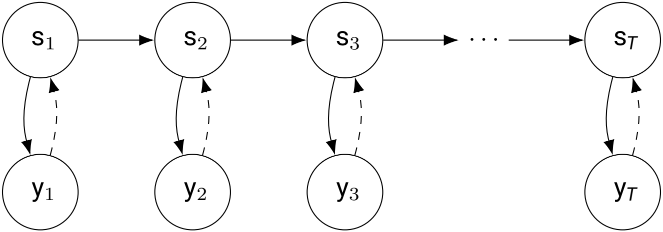
This is possible for every nonlinear dynamical system. And globalizes linearly! But the embedding g is typically infinite-dimensional…

Koopman Dynamical System (“Deterministic”)

$$
\begin{tikzcd}
s_1 \arrow[r, "F"] \arrow[d, bend right, "g"'] & s_2 \arrow[r, "F"] \arrow[d, bend right, "g"'] & s_3 \arrow[r, "F"] \arrow[d, bend right, "g"'] & \cdots \arrow[r, "F"] & s_T \arrow[d, bend right, "g"'] \\
y_1 \arrow[r, "\mathcal{K}"'] \arrow[u, dashed, bend right, "g^{-1}"'] & y_2 \arrow[r, "\mathcal{K}"'] \arrow[u, dashed, bend right, "g^{-1}"'] & y_3 \arrow[r, "\mathcal{K}"'] \arrow[u, dashed, bend right, "g^{-1}"'] & \cdots \arrow[r, "\mathcal{K}"'] & y_T \arrow[u, dashed, bend right, "g^{-1}"']
\end{tikzcd}
$$

Adopted from Brunton et al. “Koopman Invariant Subspaces and Finite Linear Representations of Nonlinear Dynamical Systems for Control”.

Linear Gaussian Dynamical System



$$
\begin{aligned}
s_{t+1} &= A s_t + v, & v &\sim \mathcal{N}(\mathbf{0}, Q) \\
y_t &= g_{\theta}(s_t) + w, & w &\sim \mathcal{N}(\mathbf{0}, R)
\end{aligned}
\iff
\begin{aligned}
s_{t+1} &\sim \mathcal{N}(A s_t, Q) \\
y_t &\sim \mathcal{N}(g_{\theta}(s_t), R)
\end{aligned}
$$

How to learn?

Goal: Estimate all of the following:

- Latent dynamics matrix A.
- Measurement function $g_{\boldsymbol{\theta}}(\cdot)$ (i.e. the parameters $\boldsymbol{\theta}$).
  Variational auto-encoder without an amortization network.
- Noise covariances Q, R.

We employ an EM-algorithm to do that.

How to learn?

Goal: Estimate all of the following:

- Latent dynamics matrix A.
- Measurement function $g_{\theta}(\cdot)$ (i.e. the parameters θ).
  Variational auto-encoder without an amortization network.
- Noise covariances Q, R.

We employ an EM-algorithm to do that.

Core Problem

Some expectations cannot be evaluated in closed form for a nonlinear $g_{\theta}(\cdot)$.

How to learn?

Goal: Estimate all of the following:

- Latent dynamics matrix A.
- Measurement function $g_{\theta}(\cdot)$ (i.e. the parameters θ).
  Variational auto-encoder without an amortization network.
- Noise covariances Q, R.

We employ an EM-algorithm to do that.

Core Problem

Some expectations cannot be evaluated in closed form for a nonlinear $g_{\theta}(\cdot)$.

- Use cubature rules for evaluating the intractable expectations.
- No closed form solution for maximizing w.r.t. function parameters θ.

How to learn?

Goal: Estimate all of the following:

- Latent dynamics matrix A.
- Measurement function $g_{\boldsymbol{\theta}}(\cdot)$ (i.e. the parameters $\boldsymbol{\theta}$).
  Variational auto-encoder without an amortization network.
- Noise covariances Q, R.

We employ an EM-algorithm to do that.

Core Problem

Some expectations cannot be evaluated in closed form for a nonlinear $g_{\boldsymbol{\theta}}(\cdot)$.

- Use cubature rules for evaluating the intractable expectations.
- No closed form solution for maximizing w.r.t. function parameters $\boldsymbol{\theta}$.

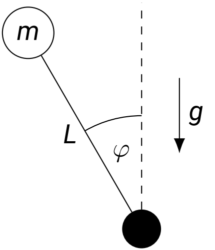
  $\longrightarrow$ Use backpropagation and gradient descent!

Results: Damped Inverted Pendulum
Dynamical System

$$\ddot{\varphi} = \frac{g}{L}\sin(\varphi) - d\dot{\varphi} \quad \longrightarrow \quad \ddot{\varphi} = \sin(\varphi) - 0.1\dot{\varphi}$$

Observations:

- Displacement φ
- Velocity $\dot{\varphi}$

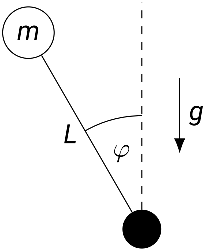


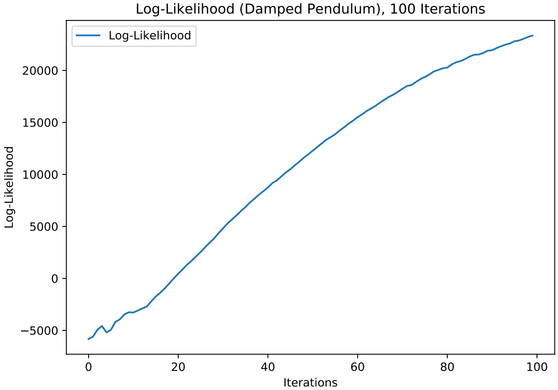
Results: Damped Inverted Pendulum Dynamical System

$$\ddot{\varphi} = \frac{g}{L}\sin(\varphi) - d\dot{\varphi} \quad \longrightarrow \quad \ddot{\varphi} = \sin(\varphi) - 0.1\dot{\varphi}$$

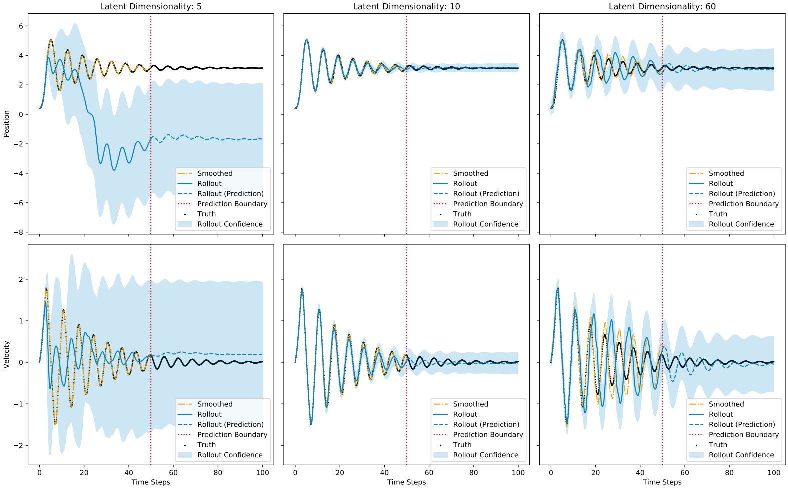
Observations:

- Displacement φ
- Velocity $\dot{\varphi}$

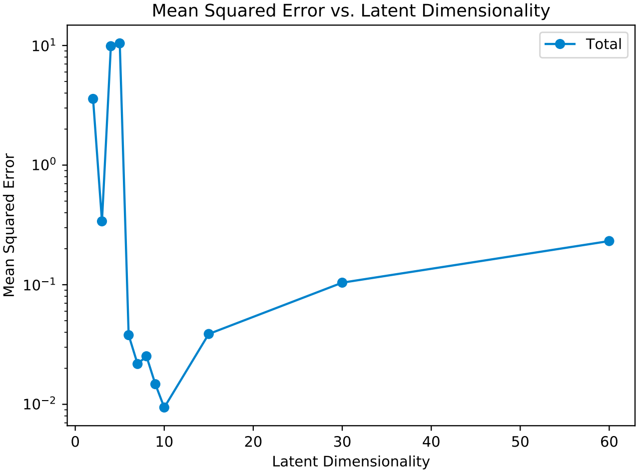




Results: Damped Inverted Pendulum Rollout/Prediction in Observation Space, 10 Latents



Results: Damped Inverted Pendulum
Different Sizes of Latent Dimension



Related Work

- Lusch et al. “Deep Learning for Universal Linear Embeddings of Nonlinear Dynamics”:
  - Use an autoencoder approach with an encoder network.
  - No probabilistic interpretation.

Related Work

- Lusch et al. “Deep Learning for Universal Linear Embeddings of Nonlinear Dynamics”:
  - Use an autoencoder approach with an encoder network.
  - No probabilistic interpretation.
- Morton et al. “Deep Variational Koopman Models: Inferring Koopman Observations for Uncertainty-Aware Dynamics Modeling and Control”:
  - Utilize six different neural networks to perform variational inference.
  - We use only one!
  - And we expect to do at least as good.

Related Work

- Lusch et al. “Deep Learning for Universal Linear Embeddings of Nonlinear Dynamics”:
  - Use an autoencoder approach with an encoder network.
  - No probabilistic interpretation.
- Morton et al. “Deep Variational Koopman Models: Inferring Koopman Observations for Uncertainty-Aware Dynamics Modeling and Control”:
  - Utilize six different neural networks to perform variational inference.
  - We use only one!
  - And we expect to do at least as good.
- Becker et al. “Recurrent Kalman Networks: Factorized Inference in High-Dimensional Deep Feature Spaces”

Related Work

- Lusch et al. “Deep Learning for Universal Linear Embeddings of Nonlinear Dynamics”:
  - Use an autoencoder approach with an encoder network.
  - No probabilistic interpretation.
- Morton et al. “Deep Variational Koopman Models: Inferring Koopman Observations for Uncertainty-Aware Dynamics Modeling and Control”:
  - Utilize six different neural networks to perform variational inference.
  - We use only one!
  - And we expect to do at least as good.
- Becker et al. “Recurrent Kalman Networks: Factorized Inference in High-Dimensional Deep Feature Spaces”
- Zhang, Vikram et al. “SOLAR: Deep Structured Representations for Model-Based Reinforcement Learning”

Hypothesis and Outlook

- Tackle the ELBO using approximate EM rather than stochastic variational inference.
- Assume we can do at least as good as Morton et al. with simpler network architectures.
- Perform control tasks on pendulum, cartpole, acrobot.

Backup Slides
Outline

Expectation Maximization

- E-Step: Calculate the expected latents and correlations using filtering/smoothing.
- M-Step: Maximize the expected log-likelihood $\mathbb{E}\big[\ln p(\mathbf{s}_{1:T}, \mathbf{y}_{1:T}) \,|\, \mathbf{y}_{1:T}\big]$.

The Expected Log-Likelihood

Markov property yields complete log-likelihood:

$$\ln p(\mathrm{s}_{1:T}, \mathrm{y}_{1:T}) = \ln p(\mathrm{s}_1) + \sum_{t=2}^{T} \ln p(\mathrm{s}_{t+1} \mid \mathrm{s}_t) + \sum_{t=1}^{T} \ln p(\mathrm{y}_t \mid \mathrm{s}_t)$$

Expectation $\mathbb{E}\left[\ln p(\mathrm{s}_{1:T}, \mathrm{y}_{1:T}) \mid \mathrm{y}_{1:T}\right]$ is based on five other expectations:

$$\hat{\mathrm{s}}_t \coloneqq \mathbb{E}\left[\mathrm{s}_t \,\middle|\, \mathrm{y}_{1:T}\right] \qquad \mathrm{P}_t \coloneqq \mathbb{E}\left[\mathrm{s}_t \mathrm{s}_t^T \,\middle|\, \mathrm{y}_{1:T}\right] \qquad \mathrm{P}_{t,t-1} \coloneqq \mathbb{E}\left[\mathrm{s}_t \mathrm{s}_{t-1}^T \,\middle|\, \mathrm{y}_{1:T}\right]$$

$$\hat{\mathrm{g}}_t \coloneqq \mathbb{E}\left[\mathrm{g}(\mathrm{s}_t) \,\middle|\, \mathrm{y}_{1:T}\right] \qquad \mathrm{G}_t \coloneqq \mathbb{E}\left[\mathrm{g}(\mathrm{s}_t)\, \mathrm{g}^T(\mathrm{s}_t) \,\middle|\, \mathrm{y}_{1:T}\right]$$

Quadrature for High Dimensions: Cubature Rules
The Spherical-Radial Cubature Rule

Approximation of an arbitrary Gaussian expectation:

$$\begin{aligned} \mathbb{E}_{\mathsf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}\big[\mathsf{f}(\mathsf{x})\big] &= \int_{\mathbb{R}^n} \mathsf{f}(\mathsf{x})\, \mathcal{N}(\mathsf{x} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma})\, \mathrm{d}\mathsf{x} \\ &\approx \frac{1}{2n} \sum_{i=1}^{2n} \mathsf{f}\Big(\sqrt{\boldsymbol{\Sigma}}\boldsymbol{\xi}_i + \boldsymbol{\mu}\Big), \quad \boldsymbol{\xi}_i = \sqrt{n}[\mathbf{1}]_i \end{aligned}$$

This finite sum can be evaluated!