2018 Intelligent Systems (**知的システム構成論**)

Methods of Learning Dynamical Systems

Apr.19, 2018

Takehisa YAIRI（**矢入健久**）
E-mail: yairi@ailab.t.u-tokyo.ac.jp

What is this lecture about ?

- Machine learning for system identification
  - In machine learning, it is called "learning dynamical systems"
- Why "learning dynamical" systems ?
  - Methods for learning static systems (e.g., supervised classification and regression) are already matured
  - Most of existing systems are dynamic in nature
  - Inference methods for dynamical systems are also matured
- Classified into three approaches:
  1. Maximum likelihood (EM-based) approach
  2. Spectral (subspace) approach
  3. (Deep) neural network approach

Schedule

- 4/19: Guidance & Introduction :
  - What is "learning dynamical systems" ?
- 5/10: Maximum Likelihood Approach 1 :
  - EM algorithm for linear dynamical systems
- 5/24: Maximum Likelihood Approach 2 :
  - Learning switching linear systems
- 5/30: Spectral Approach 1 :
  - Subspace identification
- 6/21: Spectral Approach 2 :
  - Non-linearization by kernel, Mixture model
- 7/5: Neural Network Approach :
  - Deep learning for dynamical systems

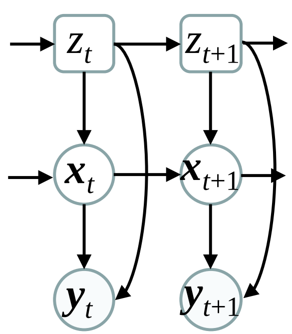
Prof. Hori will give lectures on 4/5, 4/26, 5/17, 6/14, 6/28, 7/12

Introduction to Learning Dynamical Systems

Learning Dynamical Systems

- Estimating the models of unknown dynamical systems from time-series observation data
- Known as "System Identification" in the control theory

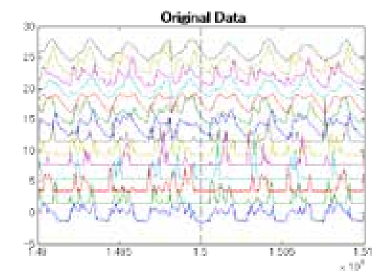
$$\begin{cases} \dfrac{d\mathbf{x}(t)}{dt} = f\left(t, \mathbf{x}(t)\right) + \mathbf{v}(t) \\ \mathbf{y}(t) = g\left(t, \mathbf{x}(t)\right) + \mathbf{w}(t) \end{cases}$$

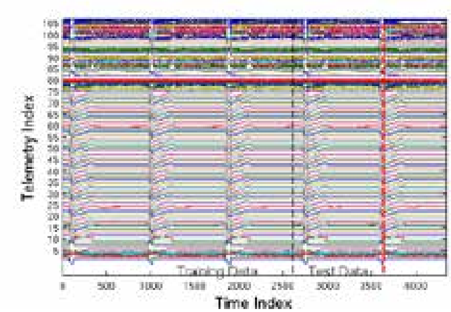


Mathematical Models

Simulation, Prediction

Time-series Observation Data





Learning Dynamical Systems / System Identification

State Space Model (SSM) in Machine Learning

- SSM is popular in machine learning, as well as in control
- Regarded as a special case of latent variable models (LVM)
  - SSM ≈ LVM with temporal structure (dynamics)
- Probabilistic representation is often used
  - Can be unified with hidden Markov models（HMM）

State transition

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{w}_t \qquad \mathbf{w}_t \sim N(\mathbf{0}, \mathbf{Q})$$

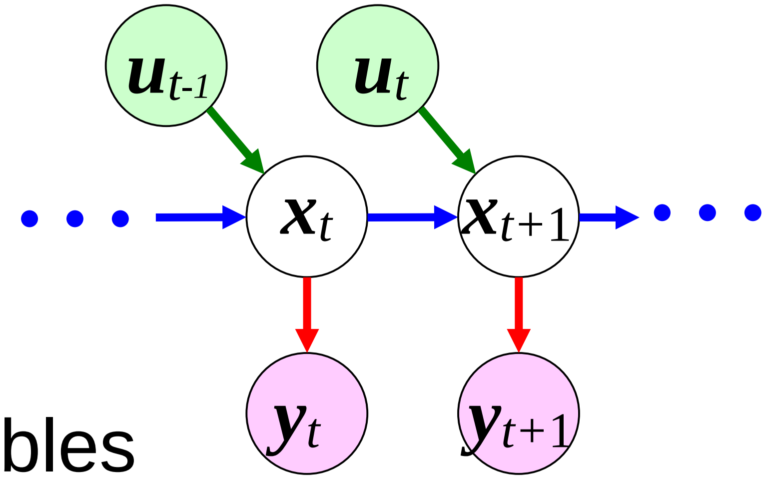
$$\Leftrightarrow \quad p(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{u}_t) = N(\mathbf{x}_{t+1} \mid f(\mathbf{x}_t, \mathbf{u}_t), \mathbf{Q})$$

Observation (Output)

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{v}_t \qquad \mathbf{v}_t \sim N(\mathbf{0}, \mathbf{R})$$

$$\Leftrightarrow \quad p(\mathbf{y}_t \mid \mathbf{x}_t) = N(\mathbf{y}_t \mid h(\mathbf{x}_t), \mathbf{R})$$

State Space Model (SSM) in Machine Learning (Cont.)



- Graph representation
  - Bayesian Networks [Pearl 88]
  - Nodes : random variables
  - Directed edges : dependencies
- Continuous and discrete variables can be mixed
  - HMM : discrete latent variables
  - Linear dynamical systems (LDS) : continuous latent variables
  - Switching LDS, Dynamic Bayesian Networks : both continuous and discrete state variables

Methods of Learning Dynamical Systems

We classifies existing methods into three categories:

1. Maximum likelihood estimation approach
   – Iteration of state estimation and model estimation
2. Spectral approach
   – Inspired by subspace identification
3. Neural network approach
   – A recent trend

Note that they are not necessarily mutually exclusive

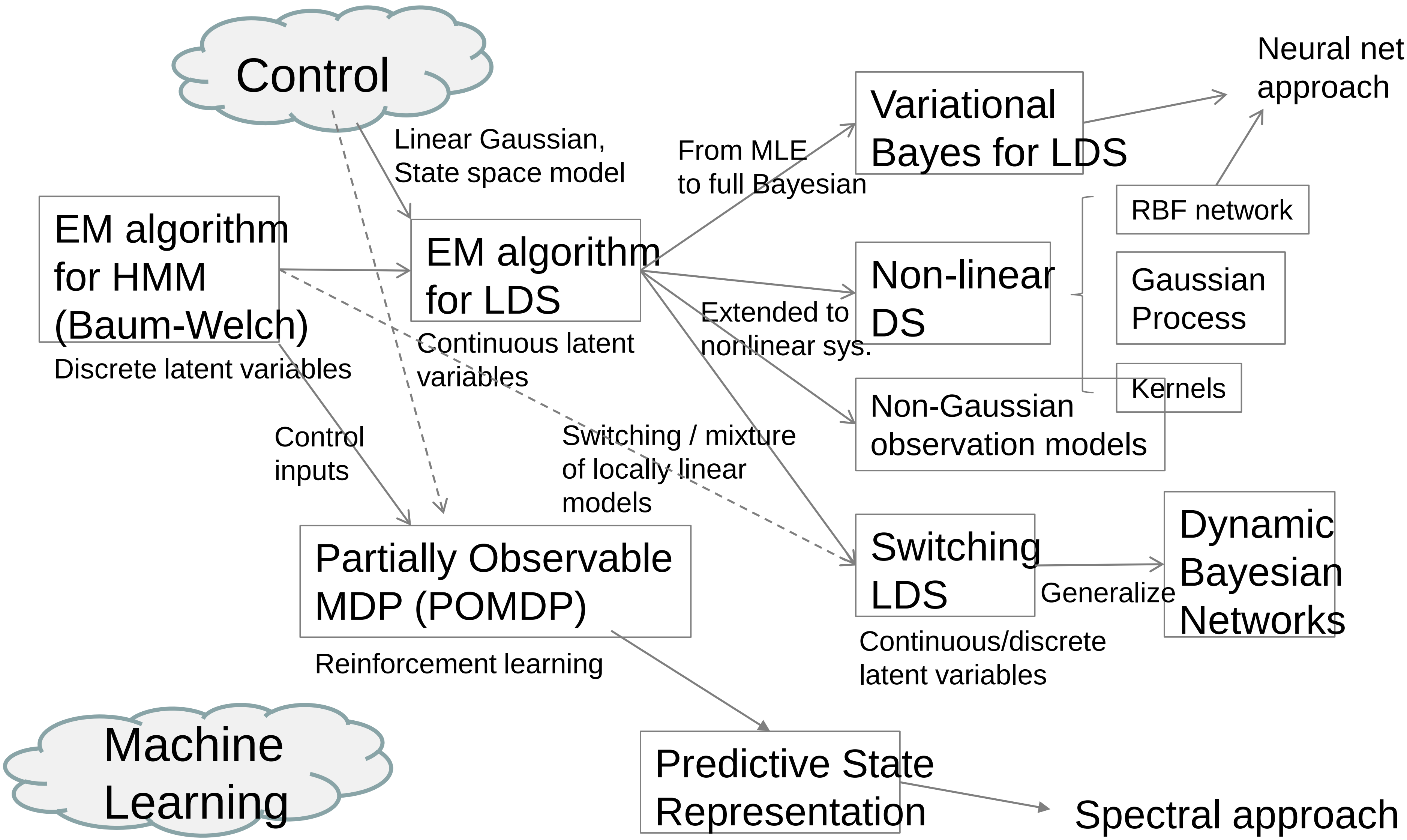
1. Maximum Likelihood Approach (In General)

- Not limited to MLE in the narrow sense
- Intended to include maximum a posteriori (MAP) and Bayesian estimation
  - MLE : $\hat{\theta}_{ML} = \arg\max_{\theta} p(\mathbf{y}_{1:T} \mid \theta) = \arg\max_{\theta} \int p(\mathbf{y}_{1:T}, \mathbf{x}_{1:T} \mid \theta) d\mathbf{x}_{1:T}$
  - MAP: $\hat{\theta}_{MAP} = \arg\max_{\theta} p(\theta \mid \mathbf{y}_{1:T}) = \arg\max_{\theta} p(\mathbf{y}_{1:T} \mid \theta) p(\theta)$
  - Bayesian: $\hat{\alpha} = \arg\max_{\alpha} p(\boldsymbol{y}_{1:T} \mid \alpha)$
    $= \arg\max_{\alpha} \left\{ \int p(\boldsymbol{y}_{1:T} \mid \theta) p(\theta \mid \alpha) d\theta \right\}$

$\boldsymbol{y}_{1:T}$: Observations
$\boldsymbol{x}_{1:T}$: States
θ : Parameters
α : Hyper param

- EM algorithm is the base method
  - In Bayesian estimation, variational inference and MCMC are also employed
- Iterations of state estimation and model estimation

Pedigree of Maximum Likelihood Approach



Expectation Maximization Algorithm

- A mandatory technique for machine learning researchers
- Maximum likelihood (or maximum a posteriori) estimation method for latent variable models
- More efficient than gradient methods
- Initialization is critical due to local minima

Given:

- Data : Y
- Initial parameter values: $\Theta^{(0)}$

Repeat until convergence

1. [E-step] Compute posterior dist. $q^*(X) = p(X \mid Y, \Theta^{(t)})$

   and $Q(\Theta \mid \Theta^{(t)}) = \mathrm{E}_{q^*(X)}[\ln p(Y, X \mid \Theta)]$

   Expected complete log-likelihood

2. [M-step] Maximize $Q(\Theta|\Theta^{(t)})$ w.r.t. Θ

   $$\Theta^{(t+1)} \leftarrow \arg\max_{\Theta} Q(\Theta \mid \Theta^{(t)})$$

3. $t \leftarrow t + 1$

We want to maximize

$$p(Y \mid \Theta) = \int p(Y, X \mid \Theta) dX$$

w.r.t. Θ but, X is also unknown

Compute $q(X) = p(X \mid Y, \Theta$

Iteration

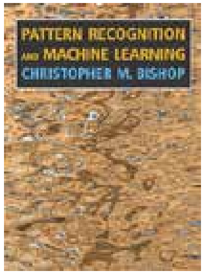
Maximize

$\mathrm{E}_{q(X)}[\ln p(Y, X \mid \Theta)]$

w.r.t. Θ

Learning Hidden Markov Model (HMM) by EM Algorithm

- Starting point of learning dynamical systems (maybe..)
  - Known as Baum-Welch algorithm
  - Included in popular machine learning textbooks such as PRML [Bishop 06]
  - HMM is popular in natural language processing, bioinformatics, activity recognition, etc. (but not in control)



- E-step : Fix model parameters, then compute the posterior distribution of discrete latent variables
- M-step : Fix the posterior of latent variables, then maximize expected likelihood w.r.t. parameters

Learning Linear Dynamical Systems by EM algorithm

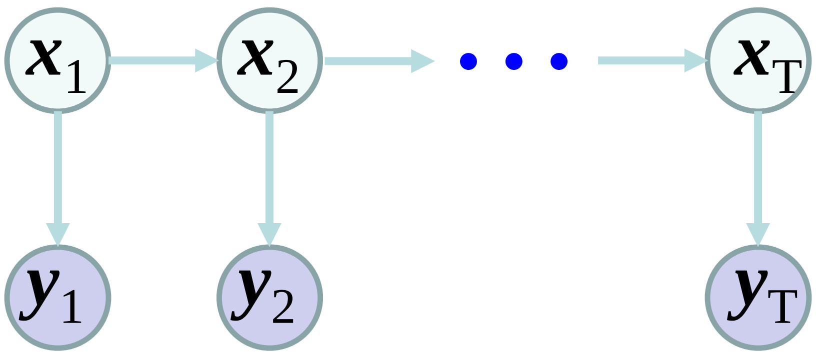
- Fundamental idea is almost the same with HMM
  - If the model is known, states can be estimated
  - If the states are known, model can be estimated
- Firstly introduced by [Ghahramani & Hinton 96]
  - Technical report, not a journal nor conference paper
- Re-introduced in PRML [Bishop 06]
- E-step = Rauch-Tung-Striebel (RTS) smoothing
  - In machine learning, it is known as Kalman smoothing
  - In fact, "Kalman filtering" is often referred as a general inference technique for state space models

EM Algorithm for Linear Dynamical Systems (Summary)

Model: $\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{x}_{t-1} + \boldsymbol{w}_t \qquad \boldsymbol{w}_t \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{Q})$

$\boldsymbol{y}_t = \boldsymbol{C}\boldsymbol{x}_t + \boldsymbol{v}_t \qquad \boldsymbol{v}_t \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{R})$

(*) Control input $\boldsymbol{u}_t$ is not considered for simplicity



Given :

– Obs. sequence : $\boldsymbol{y}_{1:\mathrm{T}}$

Find :

– System matrices : $\boldsymbol{A}$, $\boldsymbol{C}$

– Noise covariance: $\boldsymbol{Q}$, $\boldsymbol{R}$

– Posterior dist. of state sequence $\boldsymbol{x}_{1:\mathrm{T}}$: $p(\boldsymbol{x}_{1:\mathrm{T}} \mid \boldsymbol{y}_{1:\mathrm{T}})$

- Assume Gaussians : $p(\boldsymbol{x}_\mathrm{t} \mid \boldsymbol{y}_{1:\mathrm{T}}) = \mathrm{N}(\boldsymbol{x}_\mathrm{t} \mid \boldsymbol{m}_\mathrm{t}, \boldsymbol{V}_\mathrm{t})$

– Prior dist. of initial state: $p(\boldsymbol{x}_1) = \mathrm{N}(\boldsymbol{x}_1 \mid \boldsymbol{m}_0, \boldsymbol{V}_0)$

EM Algorithm for Linear Dynamical Systems（Summary）

- Initialize estimates of model parameters
- Repeat until convergence:

[E-step] Fix model parameters, then compute posterior of state sequence

For t=1:T-1Kalman filter

$$p(\boldsymbol{x}_{t+1} \mid \boldsymbol{y}_{1:t+1}) = \mathrm{N}(\boldsymbol{x}_{t+1} \mid \boldsymbol{m}_{t+1}, \boldsymbol{V}_{t+1}) \quad \begin{cases} \boldsymbol{P}_t = \boldsymbol{A}\boldsymbol{V}_t\boldsymbol{A}^T + \boldsymbol{Q} \qquad \boldsymbol{K}_{t+1} = \boldsymbol{P}_t\boldsymbol{C}^T\left(\boldsymbol{C}\boldsymbol{P}_t\boldsymbol{C}^T + \boldsymbol{R}\right)^{-1} \\ \boldsymbol{m}_{t+1} = \boldsymbol{A}\boldsymbol{m}_t + \boldsymbol{K}_{t+1}(\boldsymbol{y}_{t+1} - \boldsymbol{C}\boldsymbol{A}\boldsymbol{m}_t) \\ \boldsymbol{V}_{t+1} = (\boldsymbol{I} - \boldsymbol{K}_{t+1}\boldsymbol{C})\boldsymbol{P}_t \end{cases}$$

For t=T-1:1RTS smoother

$$p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:T}) = \mathrm{N}\left(\boldsymbol{x}_t \,\middle|\, \hat{\boldsymbol{m}}_t, \hat{\boldsymbol{V}}_t\right) \quad \begin{cases} \boldsymbol{J}_t = \boldsymbol{V}_t\boldsymbol{A}^T\boldsymbol{P}_t^{-1} \\ \hat{\boldsymbol{m}}_t = \boldsymbol{m}_t + \boldsymbol{J}_t(\hat{\boldsymbol{m}}_{t+1} - \boldsymbol{A}\boldsymbol{m}_t) \\ \hat{\boldsymbol{V}}_t = \boldsymbol{J}_t\hat{\boldsymbol{V}}_{t+1}\boldsymbol{J}_t^T + \boldsymbol{V}_t - \boldsymbol{J}_t\boldsymbol{A}\boldsymbol{V} = \boldsymbol{V}_t + \boldsymbol{J}_t\left(\hat{\boldsymbol{V}}_{t+1} - \boldsymbol{P}_t\right)\boldsymbol{J}_t^T \end{cases}$$

$$\mathrm{cov}[\boldsymbol{x}_t, \boldsymbol{x}_{t+1}] = \boldsymbol{J}_t\hat{\boldsymbol{V}}_{t+1}$$

[M-step] Fix posterior of state sequence, then update the model parameters

$$\boldsymbol{m}_0^{(t+1)} = \mathrm{E}[\boldsymbol{x}_1] = \hat{\boldsymbol{m}}_1 \qquad \boldsymbol{V}_0^{(t+1)} = \mathrm{E}\left[\boldsymbol{x}_1\boldsymbol{x}_1^T\right] - \boldsymbol{m}_0\boldsymbol{m}_0^T = \hat{\boldsymbol{V}}_1$$

$$\boldsymbol{A}^{(t+1)} = \left(\sum_{t=1}^{T-1}\mathrm{E}\left[\boldsymbol{x}_{t+1}\boldsymbol{x}_t^T\right]\right)\left(\sum_{t=1}^{T-1}\mathrm{E}\left[\boldsymbol{x}_t\boldsymbol{x}_t^T\right]\right)^{-1} \qquad \boldsymbol{C}^{(t+1)} = \left(\sum_{t=1}^{T}\boldsymbol{y}_t\,\mathrm{E}\left[\boldsymbol{x}_t^T\right]\right)\left(\sum_{t=1}^{T}\mathrm{E}\left[\boldsymbol{x}_t\boldsymbol{x}_t^T\right]\right)^{-1}$$

$$\boldsymbol{Q}^{(t+1)} = \tfrac{1}{T-1}\sum_{t=1}^{T-1}\left\{\mathrm{E}\left[\boldsymbol{x}_{t+1}\boldsymbol{x}_{t+1}^T\right] - \boldsymbol{A}^{(t+1)}\,\mathrm{E}\left[\boldsymbol{x}_t\boldsymbol{x}_{t+1}^T\right] - \mathrm{E}\left[\boldsymbol{x}_{t+1}\boldsymbol{x}_t^T\right]\boldsymbol{A}^{(t+1)^T} + \boldsymbol{A}^{(t+1)}\,\mathrm{E}\left[\boldsymbol{x}_t\boldsymbol{x}_t^T\right]\boldsymbol{A}^{(t+1)^T}\right\}$$

$$\boldsymbol{R}^{(t+1)} = \tfrac{1}{T}\sum_{t=1}^{T}\left\{\boldsymbol{y}_t\boldsymbol{y}_t^T - \boldsymbol{C}^{(t+1)}\,\mathrm{E}[\boldsymbol{x}_t]\boldsymbol{y}_t^T - \boldsymbol{y}_t\,\mathrm{E}\left[\boldsymbol{x}_t^T\right]\boldsymbol{C}^{(t+1)^T} + \boldsymbol{C}^{(t+1)}\,\mathrm{E}\left[\boldsymbol{x}_t\boldsymbol{x}_t^T\right]\boldsymbol{C}^{(t+1)^T}\right\}$$

From MLE to MAP and Bayesian Estimation

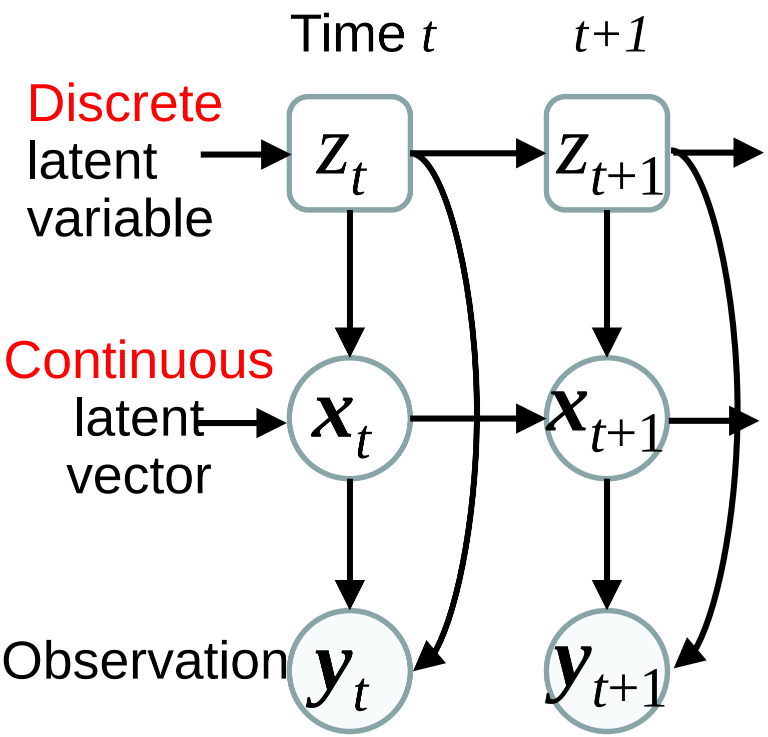
- Maximum a posteriori estimation (MAPE):
  Set prior distribution on the model parameters $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{Q}$, $\boldsymbol{R}$, then maximize the posterior probability w.r.t. the parameters
  - Reasonable when we want to use some prior knowledge or when data is not sufficient
  - EM algorithm can be used (if the prior distribution is conjugate)
- Bayesian estimation : Find the posterior distribution of the parameters $p(\boldsymbol{A},\boldsymbol{B},\boldsymbol{Q},\boldsymbol{R} \mid \boldsymbol{y}_{1:\mathrm{T}})$
  - Uncertainty of estimation is taken into consideration
  - Can beused for model selection based on marginal likelihood
  - Solved by variational Bayes[Barber06] and MCMC

Non-Gaussian Observation Model

- Advantage of maximum likelihood approach： Can be applied to "any" state space model
  - Not limited to linear Gaussian models
  - But, E-step and M-step can be more complicated..
- For example,
  - [Macke 11] [Gao 15][Park 15] :
    Dynamics of neuron activity are learned from data. Linear Gaussian state transition model + Poisson distribution observation model. EM algorithm (or variatonal EM) is used.

Switching Linear Dynamical System (SLDS)
[Murphy 98][Ghahramani&Hinton 00]

Switching among several linear models stochastically



$$\begin{cases} \mathbf{x}_{t+1} = \mathbf{A}(z_t)\mathbf{x}_t + \mathbf{w}(z_t) & \text{State transition} \\ \mathbf{y}_t = \mathbf{C}(z_t)\mathbf{x}_t + \mathbf{v}(z_t) & \text{Observation} \\ \Pr(z_{t+1} = j \mid z_t = i) = \phi_{i,j} & \text{Transition of regime (mode)} \end{cases}$$

$$\mathbf{w}(z_t) \sim N(\mathbf{0}, \mathbf{Q}(z_t))$$
$$\mathbf{v}(z_t) \sim N(\mathbf{0}, \mathbf{R}(z_t))$$

Gaussian noises

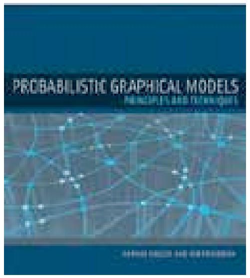
- A hybrid of HMM and linear dynamical system
- E-step is approximately computed, as analytical solution is intractable

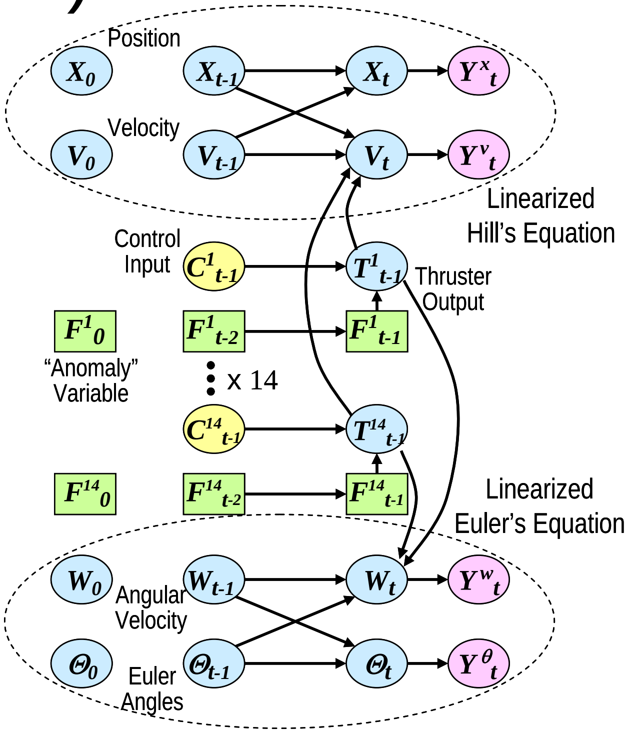
Learning of Switching Linear Dynamical Systems

- Still actively studied
- [Fox 2008] “Nonparametric Bayesian Learning of Switching Linear Dynamical Systems”, NIPS-2008
  - Dirichlet process clustering is applied
  - All parameters are estimated by MCMC
- [Chiappa 2008] “Using Bayesian Dynamical Systems for Motion Template Libraries”, NIPS-2008
  - Parameters are estimated by variational Bayes
- Applied to many purposes such as activity recognition

Dynamic Bayesian Networks

- HMM, LDS, SLDS and more complicated state space models are generalized into Dynamic Bayesian Networks (DBN)
- DBN is a special case of Bayesian networks
- Text book of DBN : [Koller 09]





Non-linear State Space Model with Radial Basis Function Network

- [Ghahramani & Roweis 1999] “Learning Nonlinear Dynamical Systems using the EM Algorithm”
- Non-linear state model f and observation model g are approximated by radial basis function networks

$$\mathbf{x}_{t+1} = f\left(\mathbf{x}_t, \mathbf{u}_t\right) + \mathbf{w}_t \qquad \text{State transition model}$$

$$\approx \sum_{i=1}^{I} h_i \underbrace{\rho_i\left(\mathbf{x}_t, \mathbf{u}_t\right)}_{\text{Gaussian RBF}} + \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t$$

- State sequence and models are iteratively estimated by EM algorithm
  - E-step : Extended Kalman (RTS) smoothing

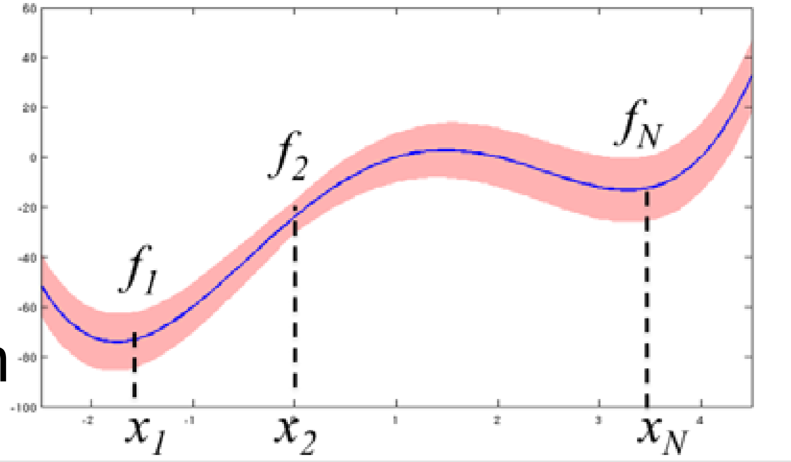
Non-linear State Space Model with Kernels

Kernel Kalman Filter (KKF) [Ralaivola 05]

- Nonlinearization by kernels (RKHS)
  - Assumed that linear dynamics and linear observation model can hold in the feature space
  - Kernel PCA is used to obtain bases in the feature space
- Learned (estimated) by EM algorithm
- Preimage is necessary
- Similar ideas of using RKHScan be seen even recently [Zhu 14]

Non-linear State Space Model with Gaussian Process

- Gaussian process regression [Rassmussen 06]
  - Supervised learning of $p(\, y \mid \boldsymbol{x} \,)$
  - Gaussian process :
    Prior distribution over a function
    $y = f(x)$
  - If $\{y_1, y_2, \ldots, y_N, y_t\}$ is Gaussian, then
    $p(y_t \mid y_1, \ldots, y_N)$ is also Gaussian
  - More compatible with probabilistic models (than SVM(SVR))
  - "Sparse" GP techniques are also available



- Gaussian Process Latent Variable Model [Lawrence 03]
  - Unsupervised learning (nonlinear dimensionality reduction)

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y} \mid \boldsymbol{x})\, p(\boldsymbol{x})\, d\boldsymbol{x}$$

$\boldsymbol{y}$: High dimensional observation

$\boldsymbol{x}$: Low dimensional latent vector

Gaussian Process State Space Models(1)

- Supervised learning of SSM
  - Assumption : State sequence $X=\{x_1,\ldots,x_N\}$ is directly observable !
    $\rightarrow$ Reduced to supervised regression problem
    $\rightarrow$ State and observation models are learned by ordinary Gaussian process regression
- GP-Bayesfilter[Ko 08], GP-ADF[Deisenroth 09]
  - $f(\boldsymbol{x}_t, \boldsymbol{u}_t)$ and $g(\boldsymbol{x}_t)$ are learned by GP regression
  - Semi-parametric
  - Sparse GP is also considered
  - Applied to system identification of blimp dynamics

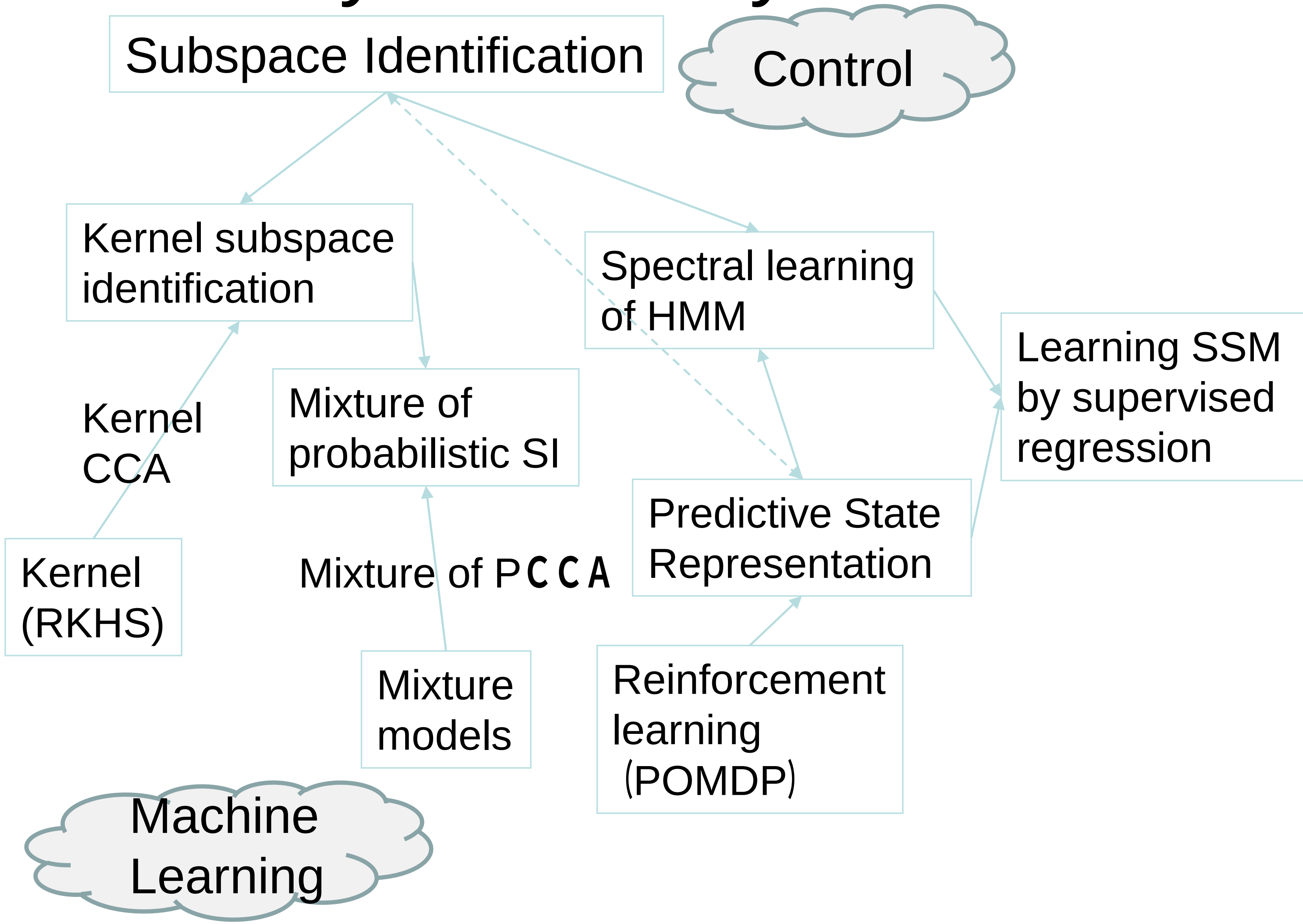
Gaussian Process State Space Models(2)

- Gaussian Process Dynamical Models [Wang 08]
  - $f(\boldsymbol{x}_t, \boldsymbol{u}_t)$ and $g(\boldsymbol{x}_t)$ are learned by GP regression
  - Marginal likelihood is maximized w.r.t. state sequence $\boldsymbol{x}_{1:T}$ and hyper parameters
  - Initialized by (ordinary) PCA
- GP Inference and Learning [Turner 10]
  - Sparse GP model using pseudo inputs
- Variational GPDS [Damianou 2011]
  - Variational Bayes inference instead of EM algorithm
- Variational GPSSM [Frigola 2013]
  - Combination of GP-based state model and parametric observation model

2. Spectral Learning of Dynamical Systems (Overview)

- Subspace identification from a machine learning perspective
- Why is it called "spectral" learning ?
  - Solved by eigen-decomposition and SVD
  - Related to manifold learning (?)
- Non-iterative algorithm and global optimum
- Less flexible than the maximum likelihood approach
- Spectral learning for HMM [Hsu 09]

Pedigree of Spectral Learning of Dynamical Systems



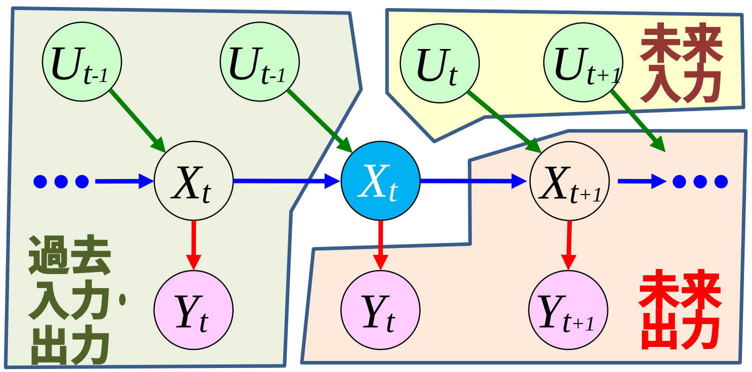
Kernel Subspace Identification [Kawahara 06]

Y. Kawahara, T. Yairi, and K. Machida, "A kernel subspace method by stochastic realization for learning nonlinear dynamical systems", NIPS-2006

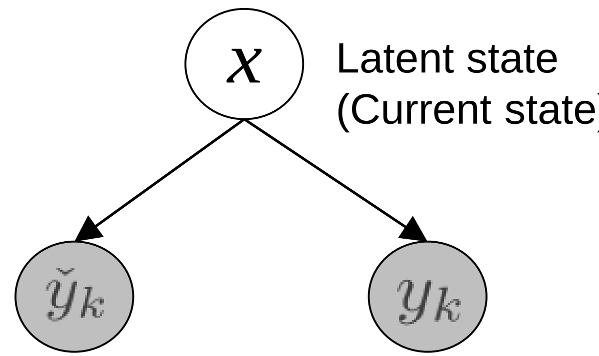
- Non-linearization of subspace identification based on canonical correlation analysis [Larimore 90][Katayama 99] by kernel (RKHS)
  - Kernel canonical correlation analysis [Akaho 01][Bach &Jordan 02]
  - KKF[Ralaivola 05] : learned by EM algorithm
- Pre-image problem is inevitable
- Pioneer work of introducing subspace identification to machine learning community

Mixture of Probabilistic Subspace Identification [Joko 11]

Subspace Identification by CCA [Larimore 90][Katayama 99]

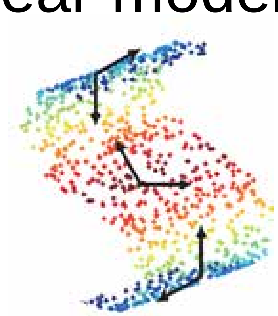


Probabilistic interpretation of CCA [Bach&Jordan 06]

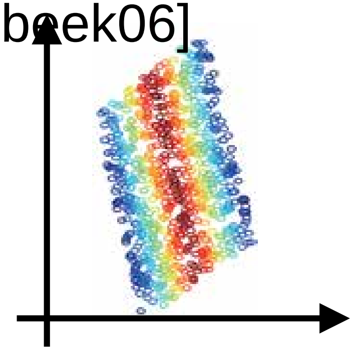


Mixture

Alignment of locally linear models [Verbeek06]



Multiple local coordinates



Global coordinates

Non-linearization of CCA subspace identification by mixture of Probabilistic CCA

- Similar to SLDS model

Spectral Learning of HMM [Hsu 09]

Daniel J. Hsu, Sham M. Kakade, and Tong Zhang, "A Spectral Algorithm for Learning Hidden Markov Models", COLT 2009.

- For a long time, EM (Baum-Welch) algorithm was believed to be the only way to learn HMM
- Inspired by subspace identification
  - Consider the canonical correlation between past and future observation
  - (Latent) state sequence and transition/output probabilities are implicitly computed
- Limitations
  - Limited to discrete observations
  - Assumption of one-step observability

Spectral Learning of HMM (Cont.)

The seminal work of [Hsu 09] was rapidly extended

- Frustration to EM algorithm

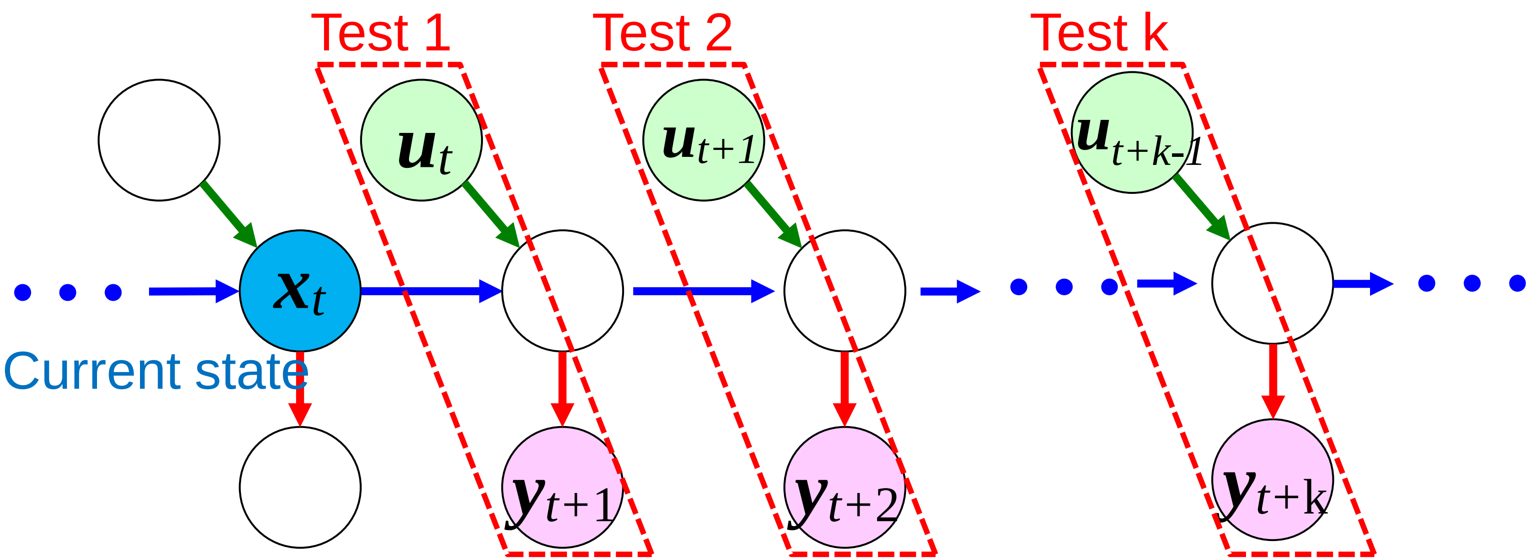
- [Siddiqi 10] Continuous observations
- [Song 10] Non-Gaussian continuous observation
- [Anandkumar 12] Generalization as a "method of moments"
- [Subakan 14] Mixture of HMM
- [Zhang 15] Latent states with tree-like structure
- [Kandasamy 16] Non-parametric observation model

Spectral Learning of Continuous State Space Models

- Spectral learning of HMM is "re-imported" to continuous state space models
  - [Buesing 12] Extention of Ho-Kalman realization algorithm to Poisson observation model
- Another idea : Use the past observation sequence, instead of estimating latent state explicitly
  - Reduced to supervised regression problems
  - [Langford 09], [Hefny 2015], [Sun 2016]
  - Predictive State Representation (PSR) [Littman 01]

Predictive State Representation (PSR)

- Originally developed as a state representation for partially observable environment [Littman 01] [Singh 04]
  - Extension of Observable operator models (OOM) [Jaeger 00]
- Instead of estimating the current state, predict a set of tests (pairs of input and output) in future
  - Predicting test results in future ≈ Guessing the current state



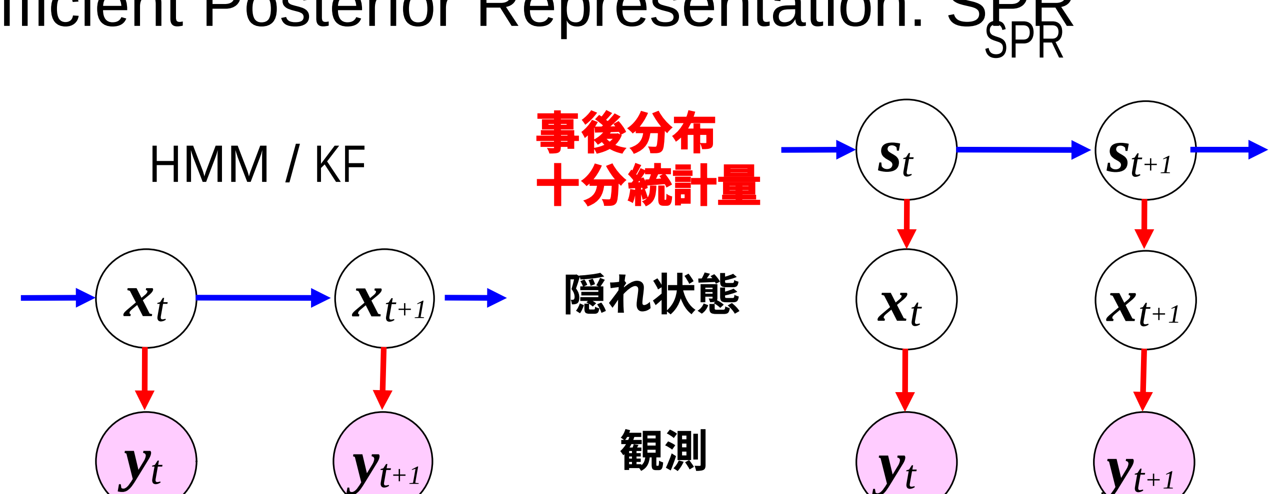
Predictive State Representation (Cont.)

- Sufficient statistic for future test results ≈ (current) state
- Predicting future test results based on past results ≈ State estimation (filtering)
- Transformed PSR [Rosencrantz 04] : Obtain a minimum set of bases necessary to predict any future test results
- Close relation to canonical variate [Akaike 75], subspace identification[Boots 09]

Langford's Method

"Learning Nonlinear Dynamic Models“, Langford , Salakhutdinov, Zhang, ICML-2009

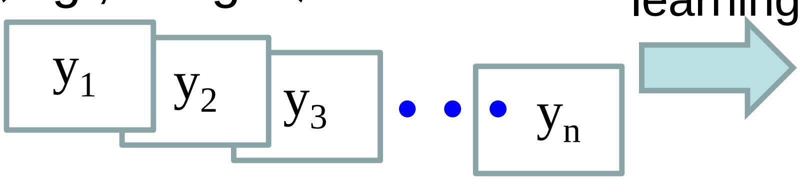
- Convert the learning of stochastic dynamical systems into deterministic supervised regression problems
- Transforms the probabilistic state $\mathbf{x}_t$ into a deterministic variable $\mathbf{s}_t$
  - Sufficient Posterior Representation: SPR



3. Neural Network Approach to Learning Dynamical Systems

- Maybe, [Roweis & Ghahramani 98] is the first
  - State space model with RBFN
- Recurrent neural networks (RNN)
- Deep learning for dynamical systems ?
- Variational autoencoder [Kingma 14]
  - Unsupervised learning of generative latent variable models

High-dimensional data
(e.g., images)



Unsupervised learning

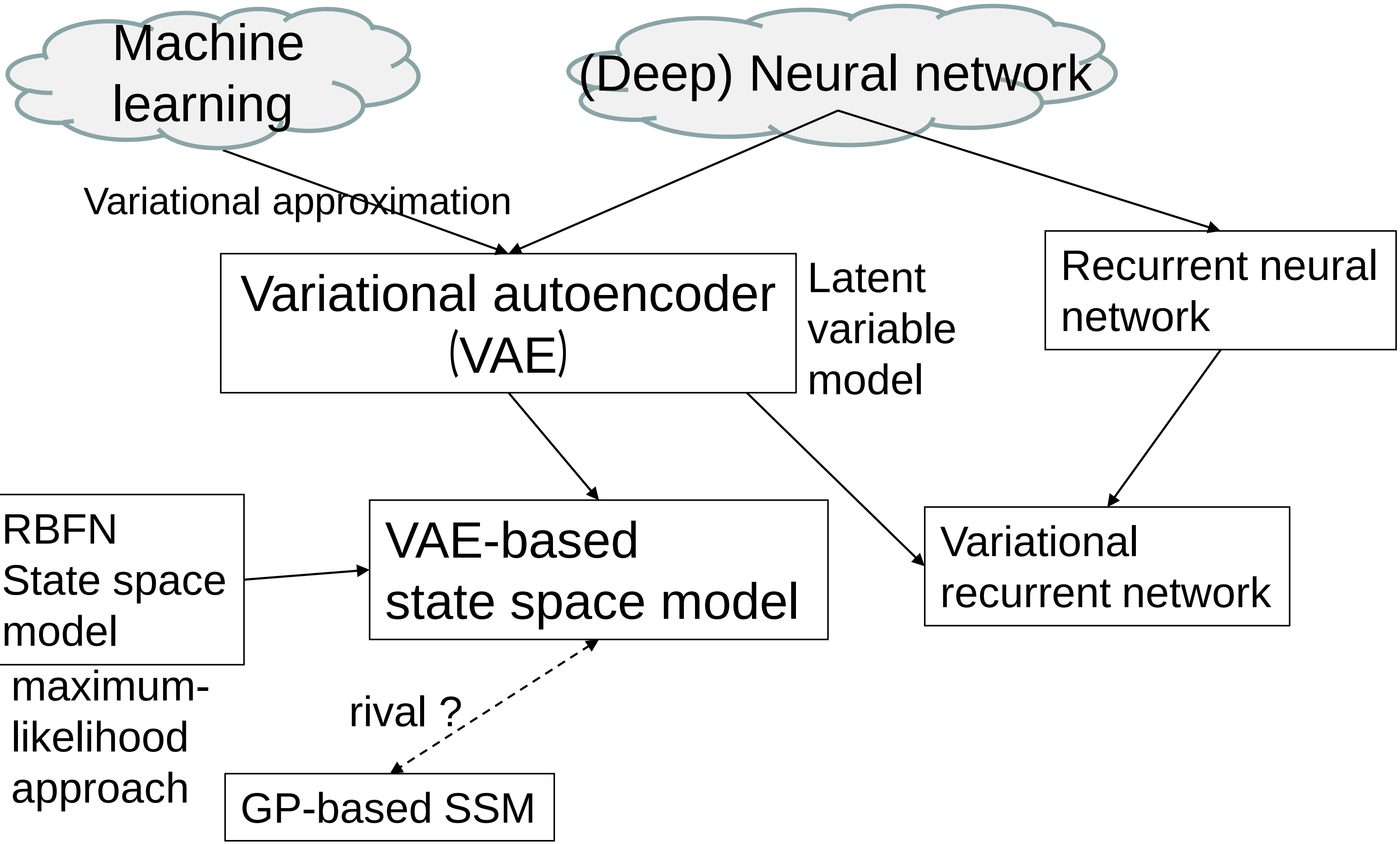
Latent variable model (e.g,VAE)

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y} \mid \boldsymbol{x})\, p(\boldsymbol{x})\, d\boldsymbol{x}$$

$\boldsymbol{y}$: high-dimensional observation

$\boldsymbol{x}$: low-dimensional latent state

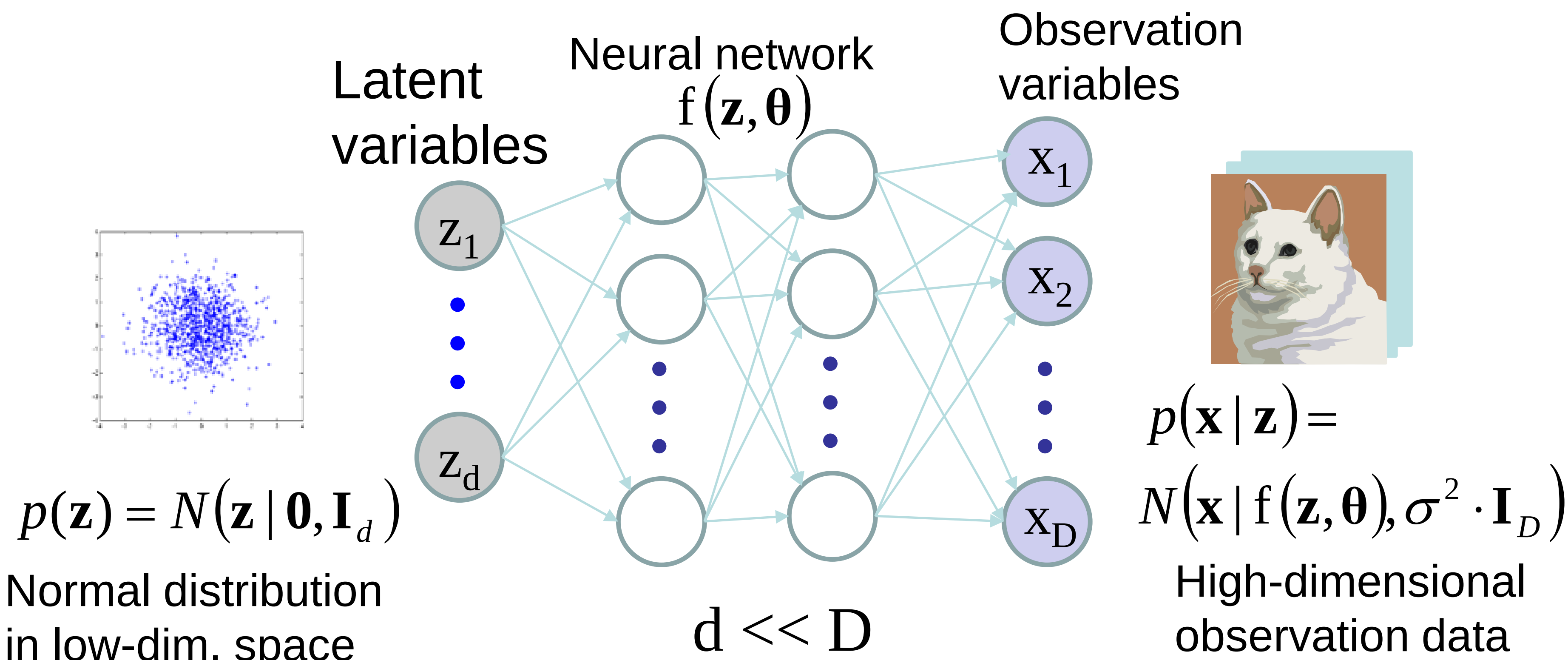
Pedigree of Neural Network Approach to Learning Dynamical Systems



Variational Autoencoder [Kingma 14]

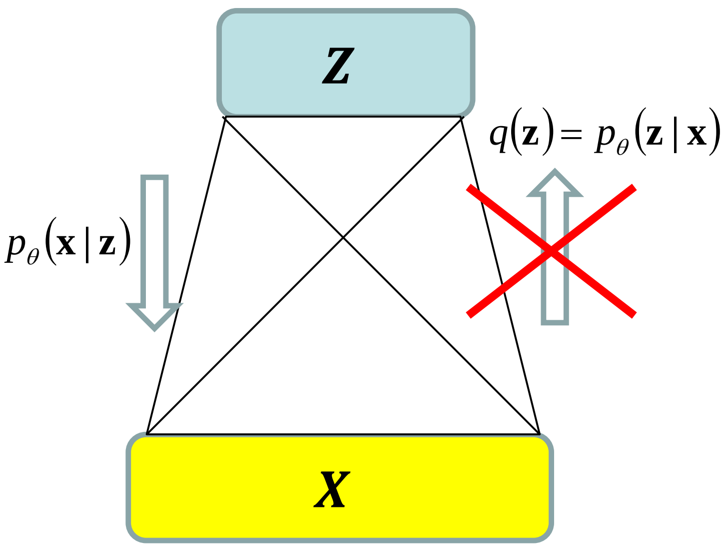
Variational Autoencoder (VAE) [Kingma 14]

- A latent variable model using neural network
  - Probabilistic, generative model
  - Powerful approximation ability of deep neural network

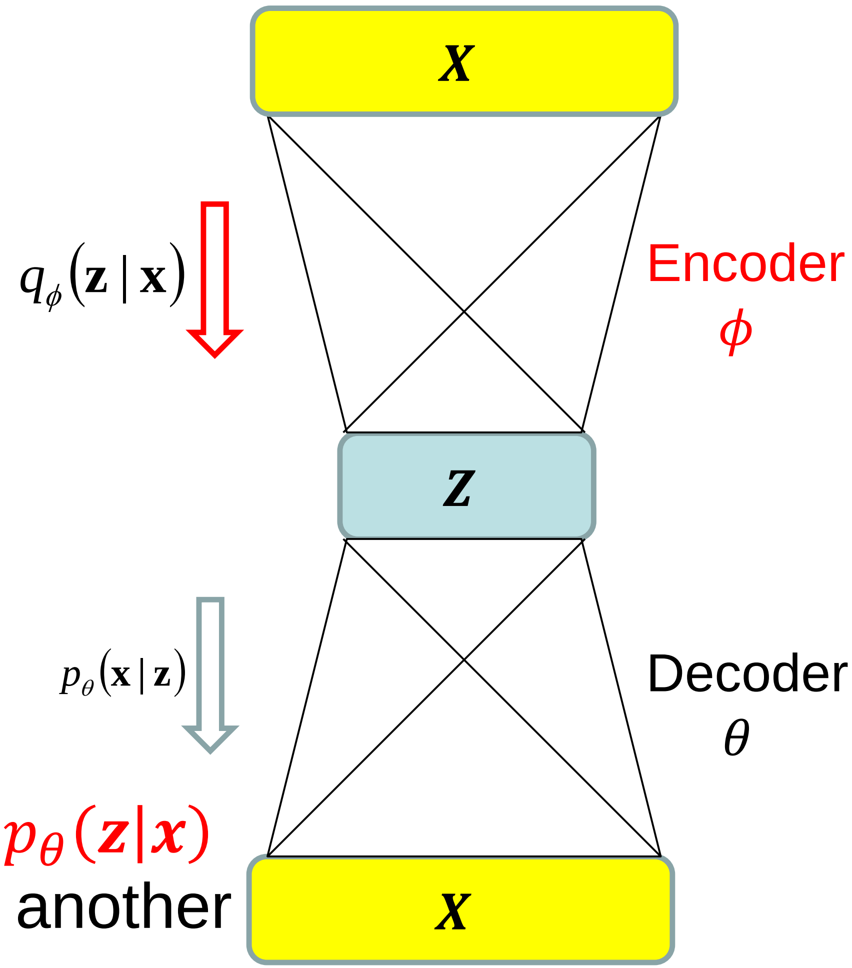


Variational Autoencoder (cont.)

EM algorithm



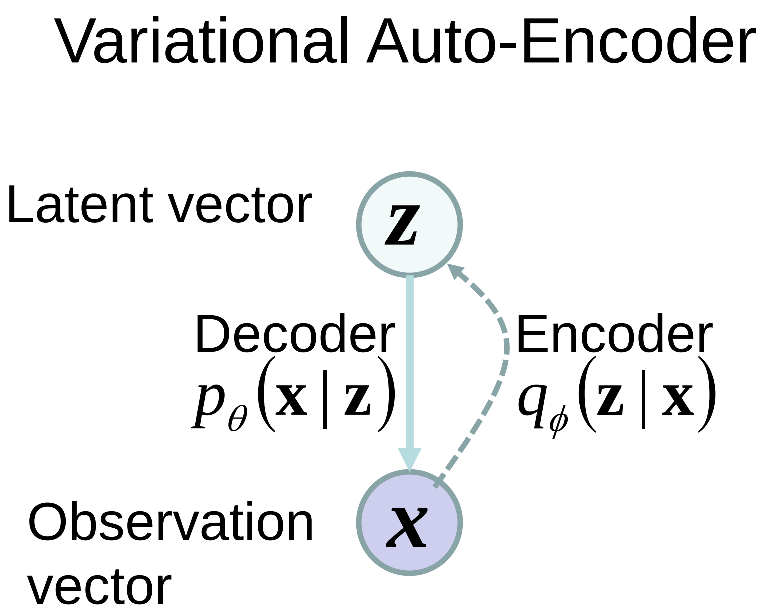
VAE (Encoder-Decoder)

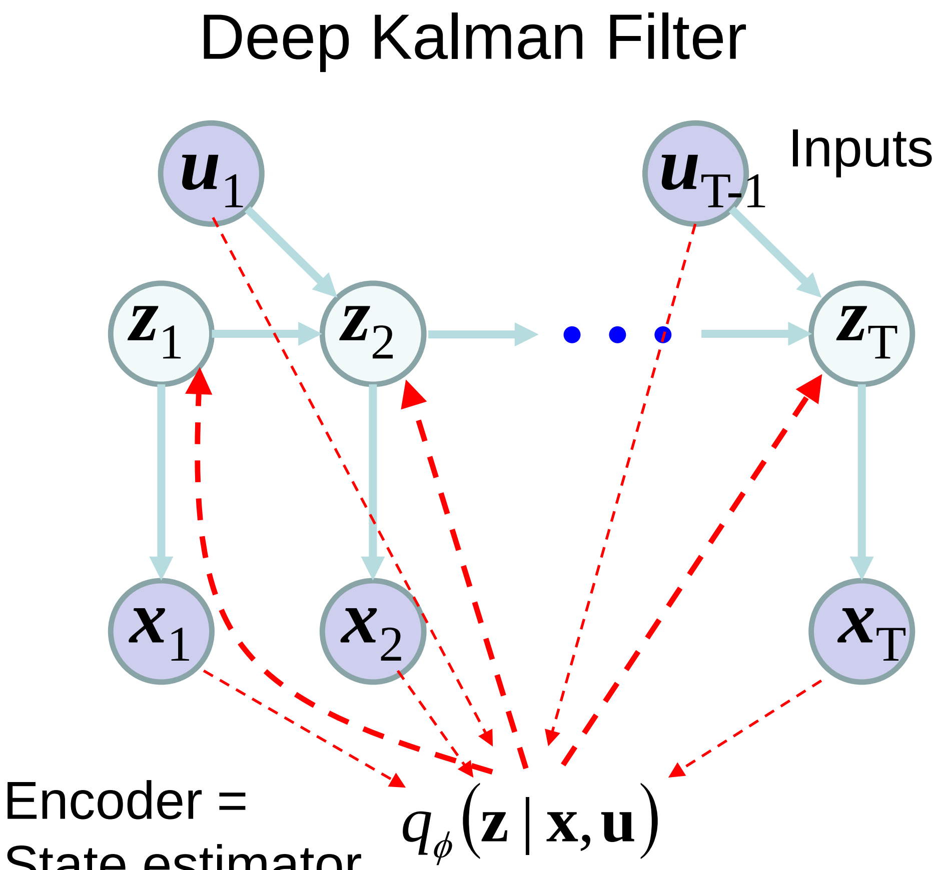


Replace the intractable posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ with its approximation $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ by another neural network

Deep Kalman Filter [Krishnan 15]

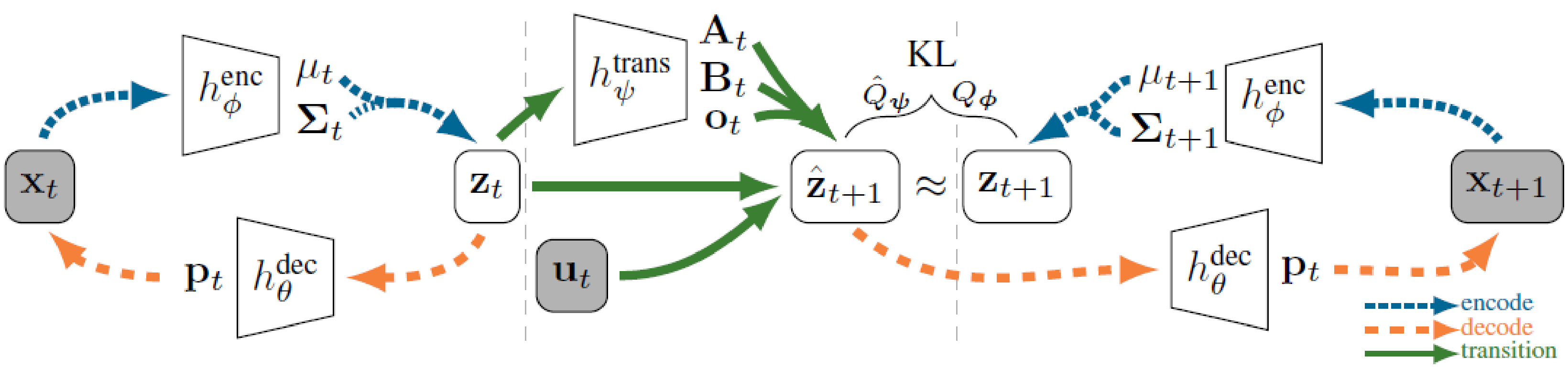
State space model with VAE





Embed to control [Watter 2015]

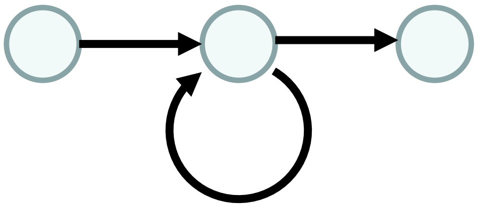
- Learning a non-linear dynamical system from a sequence of raw images
- Observation function (from latent state to observation) is modeled by VAE
- State transition is assumed to be locally linear



[Watter 2015]

Recurrent Neural Networks

- A neural network model for sequence or time-series data
- RNN has the internal state (memory)
  - Not probabilistic, but deterministic



Inspired by Variational Autoencoder

- Recurrent Latent Variable Model[Chung 15]
  - An alternative to the state space model
- Stochastic recurrent network (STORN) [Bayer 15]

Next Week

- We will begin by maximum likelihood approach
- We will derive the EM algorithm for linear dynamical systems