Learning and Inference with Dynamical Systems[1]

Jaesik Choi

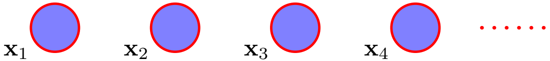
Statistical Artificial Intelligence Laboratory (SAIL)*

UNIST

*http://sail.unist.ac.kr

[1]Slides are based from [Bishop, Pattern Recognition and Machine Learning, 2007]

Introduction



- Sets of data points assumed to be independent and identically distributed (i.i.d.) in many popular models
- i.i.d. is not a reasonable assumption for sequential data
  - Measurements of time series, daily values of a currency exchange rate, acoustic features in speech recognition
  - Sequence of nucleotide base pairs along a strand of DNA, sequence of characters in an English sentence

Markov model

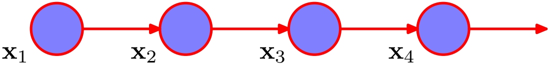
- Markov model:

$$p(x_1, \cdots, x_N) = \prod_{n=1}^{N} p(x_n | x_1, \cdots, x_{n-1})$$

- Each of the conditional distributions is independent of all previous observations except N most recent

The first-order Markov Chain

- Homogeneous Markov chain



- Joint distribution for a sequence of N observations

$$p(x_1, \cdots, x_N) = p(x_1) \prod_{n=2}^{N} p(x_n | x_{n-1})$$

- From the d-seperation property

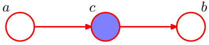
$$p(x_n | x_1, \cdots, x_{n-1}) = p(x_n | x_{n-1})$$

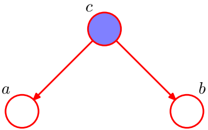
Given x_{n-1}, x_n is conditionally independent of x_1, $\cdots$, x_{n-1}.

D-separation (Pearl, 1988)

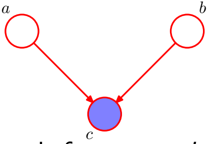
Given a directed acyclic graph, two variables a, b and a set of variables C. Any path from a to b is said to be blocked, if it includes a node such that either

- arrows on the path meet either head-to-tail or tail-to-tail at the node in the C (**inclusion**),





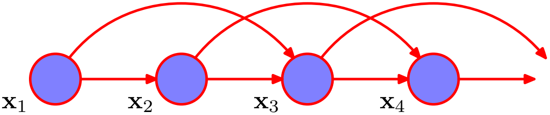
- the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, is in the set C (**exclusion**)



If every path from a to b is blocked, then, a is said to be *d-separated* from b by C, and a is conditionally independent of b given C.

A higher-order Markov chain

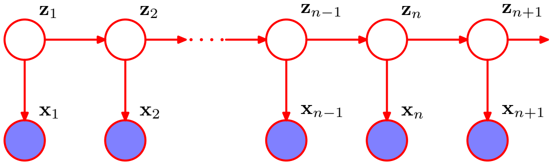
The second-order Markov chain



A higher-order Markov chain

- Observations are discrete variables having K states
- first-order: $K - 1$ parameters for each K states $\rightarrow K(K-1)$ parameters
- Mth order: $K^M(K-1)$ parameters
  E.g., $p(x|y,z)$: a second order MC with binary variables (K=2)
  $p(x=0|y{=}0,z{=}0)$, $p(x=0|y{=}0,z{=}1)$, $p(x=0|y{=}1,z{=}0)$, $p(x=0|y{=}1,z{=}1)$

Hidden Markov models (HMMs)



- z_n latent variables (discrete) (assumption: $z_{n+1} \perp\!\!\!\perp z_{n-1} | z_n$)
- x_n observed variables
- The joint distribution of the state space model

$$p(x_1, \cdots, x_N, z_1, \cdots, z_N) = p(z_1) \left[\prod_{n=2}^{N} p(z_n | z_{n-1}) \right] \prod_{n=1}^{N} p(x_n | z_n)$$

- The hidden variables make possible to represent a mixture model.

Hidden Markov Models (HMMs)

- Transition probability
  - The probability of z_n has the kth value when z_{n-1} has the jth value. Let's denote

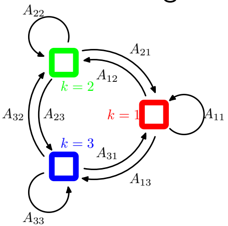
$$A_{jk} \equiv p(z_{nk} = 1 | z_{n-1,j} = 1),$$

$0 \le A_{jk} \le 1$ and $\sum_k A_{jk} = 1$.
  - Latent variables are K-dimensional binary variables.
$z_n = \{z_{n1}, z_{n2}, \cdots, z_{nK}\}$, e.g., $\{0, 1, \cdots, 0\}$ when $z_n = 2$.

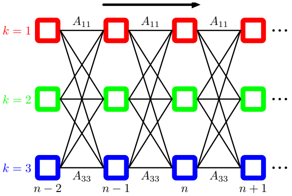
$$p(z_n | z_{n-1}, A) = \prod_{k=1}^{K} \prod_{j=1}^{K} A_{jk}^{z_{n-1,j} z_{nk}}$$

$$p(z_1 | \pi) = \prod_{k=1}^{K} \pi_k^{z_{1k}}$$

Transtion diagram



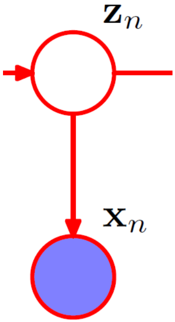
Unfolded Lattice diagram



Hidden Markov Models (HMMs)

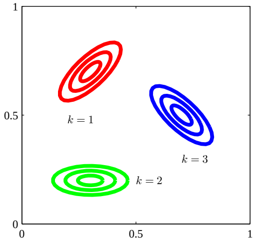
- Emission probability (observations)

$$p(x_n|z_n, \phi) = \prod_{k=1}^{K} p(x_n|\phi_k)^{z_{nk}}$$

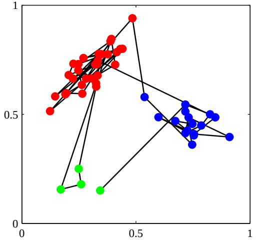


ϕ_k is the parameter of the model.

Three hidden models

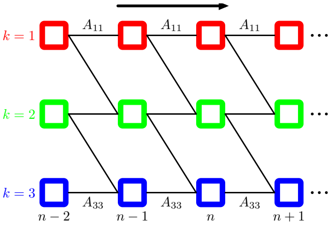


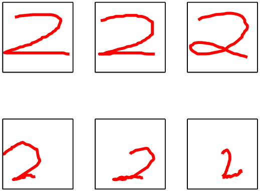
Samples when p of transition is 5%



HMM applications

- Speech recognition
- Natural language modeling
- Analysis of biological sequences
- On-line handwriting recognition (Handwritten digits)
  - Left-to-right architecture: A_{jk} of A to zero if $j > k$.
  - On-line data: each digit represented by the trajectory of the pen as a function of time





Bottom: synthetic digits from a left-to-right HMM. 16 sequences of 16 stroke (angle) directions.

Maximum likelihood for the HMM

- We have observed a data set $X = x_1, \cdots, x_N$,
- Determine the parameters of an HMM $\theta = \pi, A, \phi$
- The likelihood function is $p(X|\theta) = \sum_Z p(X, Z|\theta)$
- The maximum likelihood estimate is $\operatorname{argmax}_\theta P(X|\theta)$

Maximizing the likelihood function: EM

Expectation maximization algorithm (EM)

- Initial selection for the model parameters: θ^{old}
- E step:
  - Posterior distribution of the latent vars: $p(Z|X, \theta^{old})$
  - A log likelihood function: $p(X, Z|\theta)$

$$
\begin{aligned}
Q(\theta, \theta^{old}) &= \mathbb{E}_{p(Z|X,\theta^{old})}[\ln p(X, Z|\theta)] \\
&= \sum_{Z} p(Z|X, \theta^{old}) \ln p(X, Z|\theta)
\end{aligned}
$$

Maximizing the likelihood function: EM

E step:

$$Q(\theta, \theta^{old}) = \underbrace{\sum_{k=1}^{K} \gamma(z_{1k}) \ln \pi_k}_{\text{log of initial}} + \underbrace{\sum_{n=2}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} \xi(z_{n-1,j}, z_{nk}) \ln A_{jk}}_{\text{log of transition}}$$

$$+ \underbrace{\sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \ln p(x_n|\phi_k)}_{\text{log of emission}}$$

- The marginal posterior distribution of a latent variable γ and the joint posterior distribution of two successive latent variables ξ

$$\begin{aligned} \gamma(z_n) &= p(z_n|X, \theta^{old}) \\ \xi(z_{n-1}, z_n) &= p(z_{n-1}, z_n|X, \theta^{old}) \end{aligned}$$

Maximizing the likelihood function: EM

M step:

- Maximize $Q(\theta, \theta^{old})$ with respect to parameters $\theta = \{\pi, A, \phi\}$, treat $\gamma(z_n)$ and $\xi(z_{n-1}, z_n)$ as constant. By using Lagrange multipliers[2]

$$\pi_k = \frac{\gamma(z_{1k})}{\sum_{j=1}^{K} \gamma(z_{1j})}$$

$$A_{jk} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nk})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nl})}$$

That is, the marginals become the parameter of the categorical (multinoulli) distribution.

[2] http://cs.berkeley.edu/~stephentu/writeups/hmm-baum-welch-derivation.pdf

Lagrangian of $Q(\theta, \theta^{old})$

Lagrangian:

$$\mathcal{L}(\theta, \theta^{old}) = Q(\theta, \theta^{old}) - \lambda_\pi \left(\sum_{k=1}^{K} \pi_k - 1 \right) - \sum_{j=1}^{K} \lambda_{A_j} \left(\sum_{k=1}^{K} A_{jk} - 1 \right) - \sum_{j=1}^{K} \lambda_{p(x_n|\phi_j)} \left(\sum_{x_n} p(x_n|\phi_j) - 1 \right)$$

Lagrangian of $Q(\theta, \theta^{old})$

Partial derivative (set to 0):

$$\frac{\partial \mathcal{L}(\theta, \theta^{old})}{\partial A_{jk}} = \underbrace{\sum_{n=2}^{N} \frac{\xi(z_{n-1,j}, z_{nk})}{A_{jk}}}_{\partial \text{ of log of transition}} - \underbrace{\lambda_{A_j}}_{\partial \text{ of lagrangian term}} = 0$$

$$\Longrightarrow A_{jk} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nk})}{\lambda_{A_j}}$$

$$\frac{\partial \mathcal{L}(\theta, \theta^{old})}{\partial \lambda_{A_j}} = -\left(\sum_{k=1}^{K} A_{jk} - 1\right) = 0$$

$$\Longrightarrow \lambda_{A_j} = \sum_{k=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nk})$$

Maximizing the likelihood function: EM

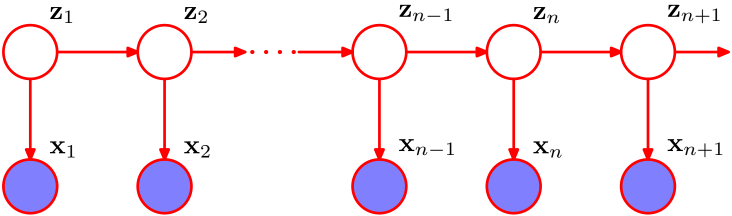
M step:

- Parameters ϕ_k independent for Gaussian emission densities $p(x|\phi_k) = \mathcal{N}(x|\mu_k, \Sigma_k)$

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

$$\sum_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

The Forward-Backward algorithm



- Two-stage message passing algorithm to compute marginals γ and ξ
- Here, we focus on alpha-beta algorithm

$$\alpha(z_n) \equiv p(x_1, \cdots, x_n, z_n) \tag{13.34}$$
$$\beta(z_n) \equiv p(x_{n+1}, \cdots, x_N | z_n) \tag{13.35}$$

Conditional Independence Properties in HMM

Use $X_{[i,j]} = (x_i, \cdots, x_j)$ as a simplied notation,

$$p(X|z_n) = p(X_{[1,n]}|z_n)p(X_{[n+1,N]}|z_n) \qquad (13.24)$$

$X_{[1,n]}$ and $X_{[n+1,N]}$ are independent given z_n

$$p(X_{[1,n-1]}|\underline{x_n}, z_n) = p(X_{[1,n-1]}|z_n)$$

$$p(X_{[1,n-1]}|z_{n-1}, \underline{z_n}) = p(X_{[1,n-1]}|z_{n-1})$$

$$p(X_{[n+1,N]}|\underline{z_n}, z_{n+1}) = p(X_{[n+1,N]}|z_{n+1})$$

$$p(X_{[n+2,N]}|z_{n+1}, \underline{x_{n+1}}) = p(X_{[n+2,N]}|z_{n+1})$$

$$p(X|z_{n-1}, z_n) = p(X_{[1,n-1]}|z_{n-1},\underline{z_n})p(x_n|\underline{z_{n-1}},z_n)p(X_{[n+1,N]}|\underline{z_{n-1}},z_n)$$

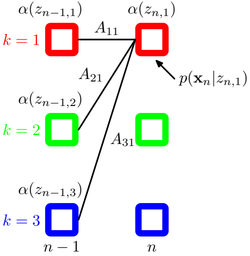
$$= p(X_{[1,n-1]}|z_{n-1})p(x_n|z_n)p(X_{[n+1,N]}|z_n)$$

$$p(x_{N+1}|\underline{X_{[1,N]}}, z_{N+1}) = p(x_{N+1}|z_{N+1})$$

$$p(z_{N+1}|z_N, \underline{X}) = p(z_{N+1}|z_N) \qquad (13.31)$$

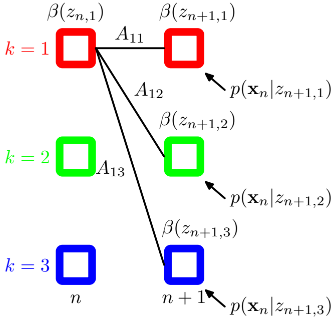
The Forward Recursion

Forward recursion for $\alpha(z_n)$



$$\alpha(z_n) = \underline{p(X_{[1,n]}|z_n)}p(z_n) = p(x_n|z_n)\underline{p(X_{[1,n-1]}|z_n)p(z_n)}$$
$$= p(x_n|z_n)\underline{p(X_{[1,n-1]}, z_n)} = p(x_n|z_n)\sum_{z_{n-1}}\underline{p(X_{[1,n-1]}, z_{n-1}, z_n)}$$
$$= p(x_n|z_n)\sum_{z_{n-1}}\underline{p(X_{[1,n-1]}|z_{n-1})}p(z_n|z_{n-1})\underline{p(z_{n-1})}$$
$$= p(x_n|z_n)\sum_{z_{n-1}}\alpha(z_{n-1})p(z_n|z_{n-1}) \qquad (13.36)$$

Backward recursion for $\beta(z_n)$



$$
\begin{aligned}
\beta(z_n) &= \sum_{z_{n+1}} \beta(z_{n+1}) p(x_{n+1}|z_{n+1}) p(z_{n+1}|z_n) \\
\beta(z_N) &= 1
\end{aligned}
$$

The Backward Recursion - Derivation

$$
\begin{aligned}
\beta(z_n) &= p(X_{[n+1,N]}|z_n) \\
&= \sum_{z_{n+1}} \underline{p(X_{[n+1,N]}, z_{n+1}|z_n)} \\
&= \sum_{z_{n+1}} p(X_{[n+1,N]}|\underline{z_n}, z_{n+1})p(z_{n+1}|z_n) \\
&= \sum_{z_{n+1}} \underline{p(X_{[n+1,N]}|z_{n+1})}p(z_{n+1}|z_n) \\
&= \sum_{z_{n+1}} \underline{p(X_{[n+2,N]}|z_{n+1})}p(x_{n+1}|z_{n+1})p(z_{n+1}|z_n).
\end{aligned}
$$

By using the definition of $\beta(z_{n+1})$,

$$
= \sum_{z_{n+1}} \beta(z_{n+1})p(x_{n+1}|z_{n+1})p(z_{n+1}|z_n). \qquad (13.38)
$$

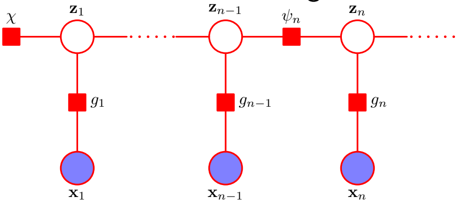
Evaluation of $\xi(z_{n-1}, z_n)$

- Using Bayes' theorem

$$
\begin{aligned}
\xi(z_{n-1}, z_n) = p(z_{n-1}, z_n|X) &= \frac{p(X, z_{n-1}, z_n)}{p(X)} \\
&= \frac{\underline{p(X|z_{n-1}, z_n)p(z_{n-1}, z_n)}}{p(X)} \\
&= \frac{\underline{p(X_{[1,n-1]}|z_{n-1})p(x_n|z_n)p(X_{[n+1,N]}|z_n)}\,\underline{p(z_n|z_{n-1})p(z_{n-1})}}{p(X)} \\
&= \frac{\underline{p(X_{[1,n-1]}|z_{n-1})p(z_{n-1})}p(x_n|z_n)\underline{p(X_{[n+1,N]}|z_n)}p(z_n|z_{n-1})}{p(X)} \\
&= \frac{\alpha(z_{n-1})p(x_n|z_n)\beta(z_n)p(z_n|z_{n-1})}{p(X)}
\end{aligned}
$$

The sum-product algorithm

- Solve the problem of finding local marginals for the hidden variables γ and ξ.
- Can be used instead of forward-backward algorithm



- Results in

$$\gamma(z_n) = \frac{\alpha(z_n)\beta(z_n)}{p(X)} \tag{13.54}$$

$$\xi(z_{n-1}, z_n) = \frac{\alpha(z_{n-1})p(x_n|z_n)p(z_n|z_{n-1})\beta(z_n)}{p(X)} \tag{13.43}$$

Scaling factors

- Used to solve forward-backward algorithm

$$\alpha(z_n) = p(x_n|z_n) \sum_{z_{n-1}} \alpha(z_{n-1}) p(z_n|z_{n-1}) \tag{13.36}$$

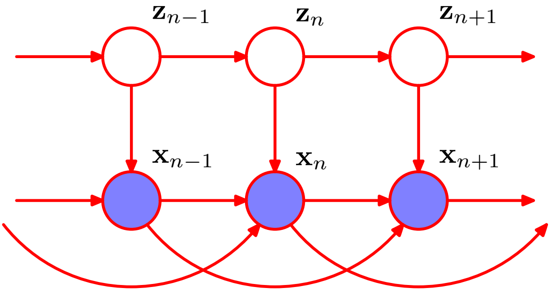
- Probabilities $p(x_n|z_n)$ and $p(z_n|z_{n-1})$ are often significantly less than unity, thus values $\alpha(z_n)$ go to zero exponentially.
- Re-scaled equations

$$\hat{\alpha}(z_n) = p(z_n|X_{[1,n]}) = \frac{\alpha(z_n)}{p(x_1, \cdots, x_n)} \tag{13.55}$$

$$\hat{\beta}(z_n) = \frac{\beta(z_n)}{p(x_{n+1}, \cdots, x_N | x_1, \cdots, x_n)}$$

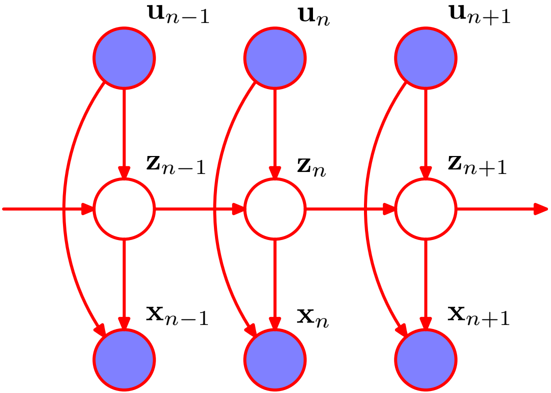
Extension I - Autoregressive HMM

- Longer-range effects could be included by adding extra link



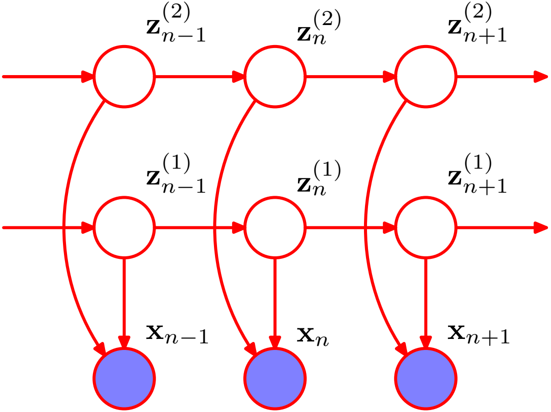
Extension II - Input-ouput HMM

- Input and output pairs could be modeled (where u_n is an input value).



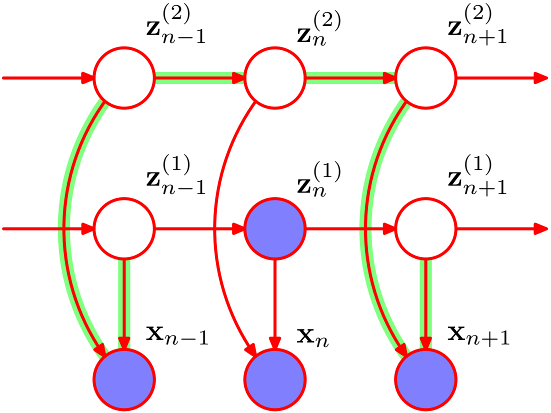
Extensions III - Factorial HMM

- The distribution of the observed variable at a given time step is conditional on the states of latent variables.



Extensions III - Factorial HMM

- Latent variables (e.g., $z_{n-1}^{(1)}$ and $z_{n+1}^{(1)}$) are not d-seperated (connected by a path given observations).



Linear Dynamical Systems (LDSs)

A Linear-Gaussian model

- The general form of algorithms for the LDS are the same as for the HMM
- Continuous latent variables
- Both observed x_n and latent z_n variables Gaussian
  - Joint distribution over all variables, marginals and conditionals are Gaussian
  - The sequence of individually most probable latent variable values is the same as the most probable latent sequence

Linear Dynamical Systems

- Transition and emission probabilities

$$p(z_n|z_{n-1}) = \mathcal{N}(z_n|Az_{n-1}, \Gamma) \quad (13.75)$$

$$p(x_n|z_n) = \mathcal{N}(x_n|Cz_n, \Sigma) \quad (13.76)$$

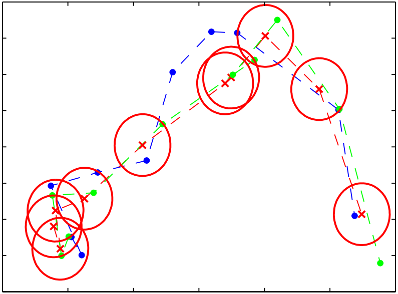
- The initial latent variable

$$p(z_1) = \mathcal{N}(z_1|\mu_0, V_0) \quad (13.77)$$

- The parameters $\theta = \{A, \Gamma, C, \Sigma, \mu_0, V_0\}$ determined using maximum likelihood through EM

Inference with Linear Dynamical Systems

- Find the marginal distributions for the latent variables conditional on the observation sequence
- Given the parameters $\theta = \{A, \Gamma, C, \Sigma, \mu_0, V_0\}$, predict the next latent state z_{n+1} and next observation x_{n+1}
- Sum-product algorithm
  - Kalman filter (forward-recursion, α message)
  - Kalman smoother (backward-recursion, β message)



blue: true position green: noisy measurements red: inferred posterior

Kalman filtering: Model

Transition model and observation model, params $\theta = \{A, \Gamma, C, \Sigma, \mu_0, P_0\}$,

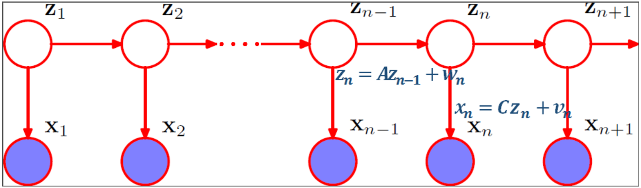
$$p(z_n|z_{n-1}) = \mathcal{N}(z_n|Az_{n-1}, \Gamma) \quad (13,75)$$

$$p(x_n|z_n) = \mathcal{N}(x_n|Cz_n, \Sigma) \quad (13.76)$$

Initial value:

$$p(z_1) = \mathcal{N}(z_1|\mu_0, P_0) \quad (13.77)$$

Linear equations expressions:



$$z_n = Az_{n-1} + w_n, \;\; x_n = Cz_n + v_n, \;\; z_1 = \mu_0 + u \quad (1)$$

with noise terms:

$$w \sim \mathcal{N}(w|0, \Gamma), \;\; v \sim \mathcal{N}(v|0, \Sigma), \;\; u \sim \mathcal{N}(u|0, P_0). \quad (2)$$

Normalized marginal distributions:

$$\hat{\alpha}(z_n) = \mathcal{N}(z_n|\mu_n, V_n). \tag{13.84}$$

Recursion equations:

$$c_n\hat{\alpha}(z_n) = p(x_n|z_n) \int \hat{\alpha}(z_{n-1})p(z_n|z_{n-1})dz_{n-1} \tag{13.85}$$

$$c_n\mathcal{N}(z_n|\mu_n, V_n) = \mathcal{N}(x_n|Cz_n, \Sigma)\underline{\int \mathcal{N}(z_{n-1}|\mu_{n-1}, V_{n-1})\mathcal{N}(z_n|Az_{n-1}, \Gamma)dz_{n-1}}$$
$$= \mathcal{N}(x_n|Cz_n, \Sigma)\mathcal{N}(z_n|A\mu_{n-1}, P_{n-1}) \tag{13.87}$$

where $P_{n-1} = AV_{n-1}A^T + \Gamma$.

Kalman filtering: Derivation

$$c_n \mathcal{N}(z_n|\mu_n, V_n) = \mathcal{N}(x_n|Cz_n, \Sigma)\mathcal{N}(z_n|A\mu_{n-1}, P_{n-1})$$

where $P_{n-1} = AV_{n-1}A^T + \Gamma$.
Solve the above equations,

$$\mu_n = A\mu_{n-1} + K_n(x_n - CA\mu_{n-1}) \tag{13.89}$$

$$V_n = (I - K_n C)P_{n-1} \tag{13.90}$$

$$c_n = \mathcal{N}(x_n|CA\mu_{n-1}, CP_{n-1}C^T + \Sigma). \tag{13.91}$$

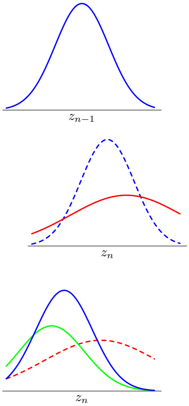
where K_n is the *Kalman gain matrix*:

$$K_n = P_{n-1}C^T(CP_{n-1}C^T + \Sigma)^{-1}$$

[Important] the exponent of a Gaussian distribution is represented by

$$\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu + constant = -\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) \tag{2.71}$$

Kalman filtering: Illustration



Before filtering

$$p(z_{n-1}|X_{[1,n-1]})$$

Diffusion by transition

$$p(z_n|X_{[1,n-1]}) = \underline{p(z_n|z_{n-1})}p(z_{n-1}|X_{[1,n-1]})$$

Sifted and Narrowed by an observation

$$p(z_n|X_{[1,n]}) = \underline{p(x_n|z_n)}p(z_n|X_{[1,n-1]})$$

Learning in Linear Dynamical Systems

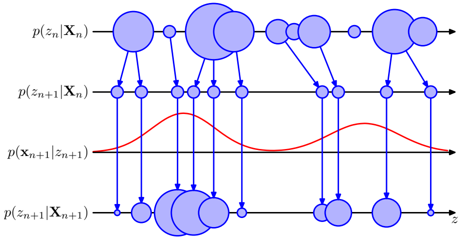
- Determine $\theta = \{A, \Gamma, C, \Sigma, \mu_0, V_0\}$ using maximum likelihood
- Expectation maximization
  - E step: $Q(\theta, \theta^{old}) - E_{Z|\theta^{old}}[\ln p(X, Z|\theta)]$
  - M step: Maximize with respect to the components of θ

Extensions of LDSs

- The marginal distribution of the observed variables is Gaussian
  use Gaussian mixture as the initial distribution for z_1
- Make Gaussian approximation by linearizing around the mean of the predicted distribution
  Extended Kalman filter
- Combining the HMM with a set of linear dynamical systems
  Switching state space model

Particle filters

Sampling methods

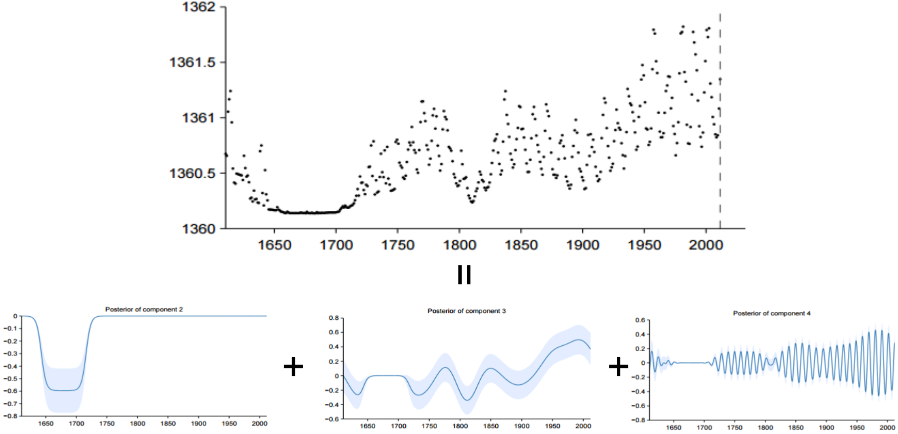


- Needed for dynamical systems which do not have a linear-Gaussian
- Sampling-importance-resampling formalism
  a sequential Monte Carlo as the particle filter
- Particle filter algorithm:
  At time step n
  - obtained a set of samples and weights
  - observe x_{n+1}
  - evaluate samples and weights for time step $n+1$

Recent Advances in Dynamical Systems

Nonparametric Regression by Gaussian Processes

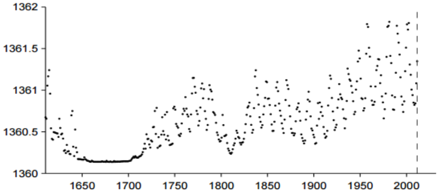
- Automatic Bayesian Covariance Discovery (ABCD, The Automatic Statistician)



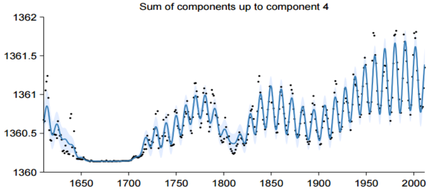
The Automatic Statistician, http://www.**automaticstatistician**.com), 2013

Nonparametric Regression by Gaussian Processes

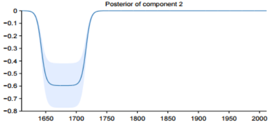
- Automatic Bayesian Covariance Discovery (ABCD, The Automatic Statistician)



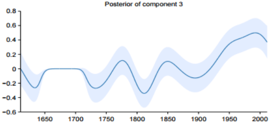
≈



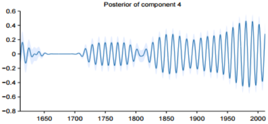
=



+

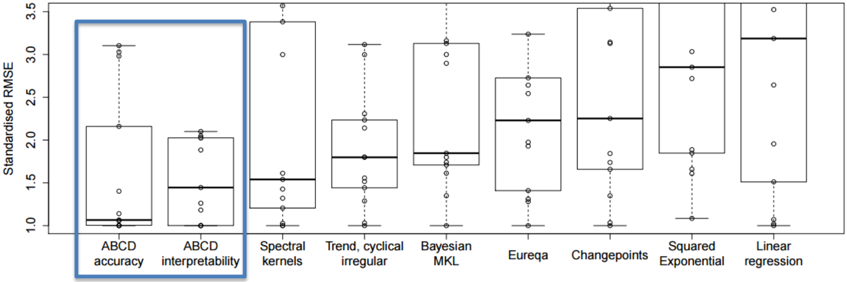


+



Nonparametric Regression by Gaussian Processes

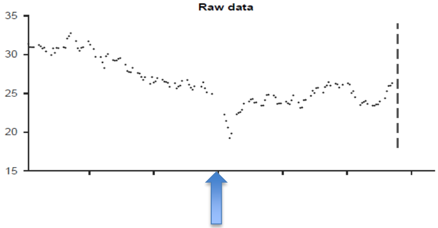
- Automatic Bayesian Covariance Discovery (ABCD, The Automatic Statistician)



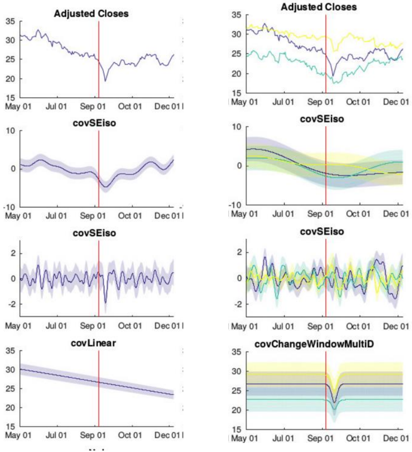
13 raw regression datasets

Nonparametric Regression by Gaussian Processes

- Relational Automatic Bayesian Covariance Discovery with Multiple datasets



The 911 attacks



ABCD (2013,2104) R-ABCD (ours, 2015)

http://sail.unist.ac.kr/yunseong/

Nonparametric Regression by Gaussian Processes

An automatic report for the dataset : GE Relational version

The Automatic Statistician

Abstract

This report was produced by the Automatic Bayesian
(ABCD) algorithm.

1 Executive summary

The raw data and full model posterior with extrapolations are showr

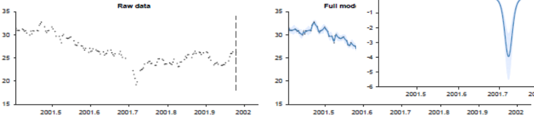
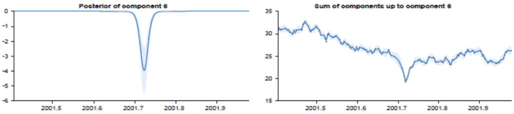


Figure 1: Raw data (left) and model posterior with extrapolation (right)

2.6 Component 6 : A constant. This function applies from 12 Sep 2001 until 15 Sep 2001

This component is constant. This component applies from 12 Sep 2001 until 15 Sep 2001.

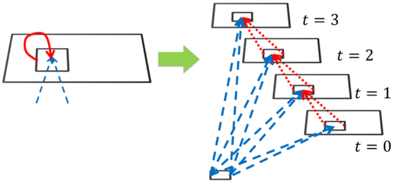
This component explains 100.0% of the residual variance; this increases the total variance explained from 95.2% to 100.0%. The addition of this component increases the cross validated MAE by 0.67% from 0.87 to 0.87. This component explains residual variance but does not improve MAE which suggests that this component describes very short term patterns, uncorrelated noise or is an artefact of the model or search procedure.



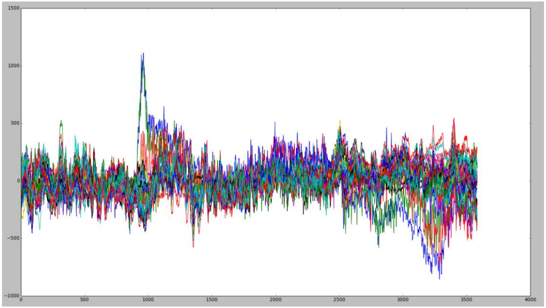
Recurrent Convolutional Neural Network for EEG analysis

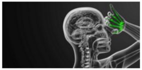
RNN $\mathbf{x}(t) = \boldsymbol{\sigma}\left(\mathbf{W}^{in}\mathbf{u}(t) + \mathbf{W}^{rec}\mathbf{x}(t-1)\right) + \mathbf{b})$

RCL $x_{ijk}(t) = \boldsymbol{\sigma}\left((\mathbf{w}_k^{in})^{\mathrm{T}}\mathbf{u}^{(i,j)}(t) + (\mathbf{w}_k^{rec})^{\mathrm{T}}\mathbf{x}^{(i,j)}(t-1) + b_k\right)$



One chunk: Data: 3584,32



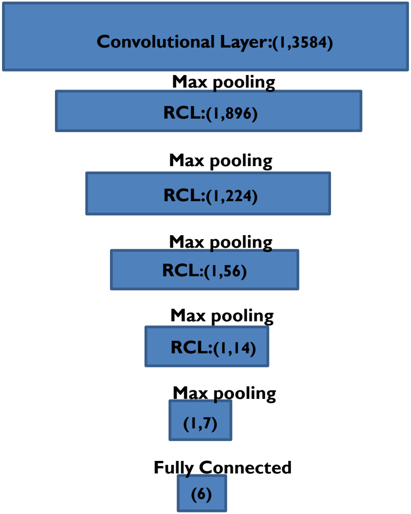


Completed • $10,000 • 379 teams

Grasp-and-Lift EEG Detection

Mon 29 Jun 2015 – Mon 31 Aug 2015 (4 months ago)

Applying RCL



Layer type	Size	Output shape
Convolutional	256 1×9 filters	(64, 256, 1, 3584)
Max pooling	Pool size 4, stride 4	(64, 256, 1, 896)
RCL	256 1×1 feed-forward filters, 256 1×9 filters, 3 iterations	(64, 256, 1, 896)
Max pooling	Pool size 4, stride 4	(64, 256, 1, 224)
RCL	256 1×1 feed-forward filters, 256 1×9 filters, 3 iterations	(64, 256, 1, 224)
Max pooling	Pool size 4, stride 4	(64, 256, 1, 56)
RCL	256 1×1 feed-forward filters, 256 1×9 filters, 3 iterations	(64, 256, 1, 56)
Max pooling	Pool size 4, stride 4	(64, 256, 1, 14)
RCL	256 1×1 feed-forward filters, 256 1×9 filters, 3 iterations	(64, 256, 1, 14)
Max pooling	Pool size 2, stride 2	(64, 256, 1, 7)
Fully connected	1792×6	(64, 6)

97.687%

Thank you!

http://sail.unist.ac.kr

jaesik@unist.ac.kr