Predictive State Recurrent Neural Networks

Downey, Hefny, Li, Boots, Gordon

Arthur Gretton's notes

September 19, 2017

Summary

Tasks

- Doing *filtering* (predicting future observations, given one new observation, and past history) or *prediction* (as above, but using only past history)

Why should we care?

- "We outperform several popular alternative approaches to modeling dynamical systems [on four datasets]"
- Better than LSTMs and GRU

Predictive state representations

A predictive state representation is made up of:

- Observations $o_1, \ldots, o_t, \ldots, o_T$
- History: $h_t = h(o_{1:t-1})$, a *vector of features* of the past observations
- Future: $f_t = f(o_{t:t+k-1})$, a vector of features of future observations

The predictive state

$$q_t = E(f_t | h_t)$$

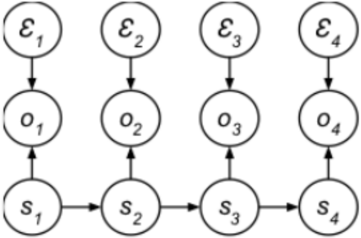
which determines

$$P(o_{t:t+k-1} | o_{1:t-1})$$

(eg a mean embedding - expected random Fourier features in the paper).

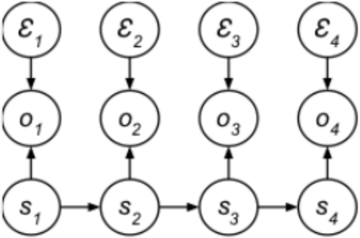
Aside: relation to instrumental variables

Can we regress from $f_t = f(o_{t:t+k-1})$ to $f_{t+1} = f(o_{t+1:t+k})$? Problem: due to window overlap, the noise variables for input and output are correlated $\rightarrow$ this introduces bias.



Aside: relation to instrumental variables

Can we regress from $f_t = f(o_{t:t+k-1})$ to $f_{t+1} = f(o_{t+1:t+k})$? Problem: due to window overlap, the noise variables for input and output are correlated $\rightarrow$ this introduces bias.



A solution: condition on instrumental variables that are correlated with input but not noise.

- Here, instrumental variables are history features $h_t = h(o_{1:t-1})$, uncorrelated with noise $\epsilon_{t:t+k}$.

PSRs with random Fourier features

Task: predict $q_{t+1} = E(f_{t+1}|h_{t+1})$ given q_t and o_t (this is an earlier paper: Supervised Learning from Dynamical Systems learning)

PSRs with random Fourier features

Task: predict $q_{t+1} = E(f_{t+1}|h_{t+1})$ given q_t and o_t (this is an earlier paper: Supervised Learning from Dynamical Systems learning)

First simpler task: predict $q_{t+1} = E(f_{t+1}|h_{t+1})$ from o_t and h_t using kernel Bayes rule.

$$\begin{aligned} q_{t+1} &= E(f_{t+1}|o_t, h_t) \\ &= C_{f_{t+1},o_t|h_t} C^{-1}_{o_t,o_t|h_t} o_t \end{aligned}$$

where

$$\begin{aligned} C_{f_{t+1},o_t|h_t} &= C_{(f_{t+1},o_t)h_t} C^{-1}_{h_t,h_t} h_t \\ C_{o_t,o_t|h_t} &= C_{(o_t,o_t)h_t} C^{-1}_{h_t,h_t} h_t \end{aligned}$$

PSRs with random Fourier features

Task: predict $q_{t+1} = E(f_{t+1}|h_{t+1})$ given q_t and o_t (this is an earlier paper: Supervised Learning from Dynamical Systems learning)

First simpler task: predict $q_{t+1} = E(f_{t+1}|h_{t+1})$ from o_t and h_t using kernel Bayes rule.

$$\begin{aligned} q_{t+1} &= E(f_{t+1}|o_t, h_t) \\ &= C_{f_{t+1},o_t|h_t} C^{-1}_{o_t,o_t|h_t} o_t \end{aligned}$$

where

$$\begin{aligned} C_{f_{t+1},o_t|h_t} &= C_{(f_{t+1},o_t)h_t} C^{-1}_{h_t,h_t} h_t \\ C_{o_t,o_t|h_t} &= C_{(o_t,o_t)h_t} C^{-1}_{h_t,h_t} h_t \end{aligned}$$

Problem: we want to condition on (and update) q_t, not condition on h_t.

PSRs with random Fourier features

Task: predict $q_{t+1} = E(f_{t+1}|h_{t+1})$ given q_t and o_t.

$$\begin{aligned} q_t &= E(f_t|h_t) \\ &= C_{f_t,h_t} C_{h_t,h_t}^{-1} h_t \end{aligned}$$

and so

$$C_{h_t,h_t}^{-1} h_t = C_{f_t,h_t}^{\dagger} q_t$$

(note pseudoinverse).

A simpler architecture

A "joint density" model (rather than conditional) with ℓ_2 normalisation,

$$q_{t+1} = \frac{W \times_2 o_t \times_3 q_t + b}{\|W \times_2 o_t \times_3 q_t + b\|_2}$$

Still multiplicatively integrates information from o_t and q_t. ("a commonly made simplification in the systems literature, and has been shown to work well in practice")

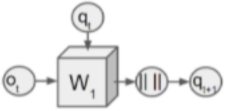
A simpler architecture

A "joint density" model (rather than conditional) with ℓ_2 normalisation,

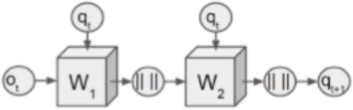
$$q_{t+1} = \frac{W \times_2 o_t \times_3 q_t + b}{\|W \times_2 o_t \times_3 q_t + b\|_2}$$

Still multiplicatively integrates information from o_t and q_t. ("a commonly made simplification in the systems literature, and has been shown to work well in practice")

Multilayer extension:



(a) Single Layer PSRNN



(b) Multilayer PSRNN

Use estimated states in place of observations.

Why chain on observation, not state? Consisent with

- LSTMs/GRU
- normalised PSRs "where observation passed through two layers"

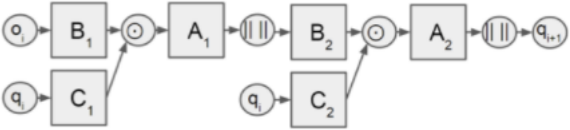
A factorised representation

Assume we have the CP decomposition for W,

$$W = \sum_{i=1}^{n} a \otimes b \otimes c$$

Then

$$\begin{aligned} q_{t+1} &= W \times_2 o_t \times_3 q_t + b \\ &= A^\top (Bo_t \odot Cq_t) + b \end{aligned}$$



A factorised representation

Assume we have the CP decomposition for W,

$$W = \sum_{i=1}^{n} a \otimes b \otimes c$$

Then

$$\begin{aligned} q_{t+1} &= W \times_2 o_t \times_3 q_t + b \\ &= A^\top (Bo_t \odot Cq_t) + b \end{aligned}$$

This shows a gating effect:

$$[q_{t+1}]_i = \sum_j A_{ji} \left(\sum_k B_{jk}[o_t]_k \odot \sum_l C_{jl}[q_t]_l \right) + b$$

So q_t contributes to q_{t+1} only if $\sum_k B_{jk}[o_t]_k$ is non-zero.

Experiments

Yarin Gal's experiment comment:
"I would take the new paper's results with a grain of salt... the experiments they have are non-standard (I've never seen that setup for PTB (Penn Tree Bank) for example; there is a standard train / test split which they ignore most likely because the method cannot scale to the full data?)"