Bayesian Multi-task Learning for Dynamic Time Series Prediction

Dr. Rohitash Chandra

1. Centre for Translational Data Science
http://sydney.edu.au/data-science
2. School of Geosciences

International Joint Conference on Neural Networks
Rio, Brazil

July 7, 2018

Overview

Background

- Dynamic time series prediction refers to the ability of a model to given a prediction given a set of input values that about past behaviour of the time series.
- Due to knowledge representation as modules, the system should be able to make decisions with some degree of uncertainty even if some of the modules are damaged or missing.
- Such motivations come from biological neural systems, i.e due to modular knowledge representation, one is able to see even if one eye is damaged.

Background and Motivation

- Multi-task learning employs shared representation knowledge for learning multiple instances from the same problem. In the case of time series, multi-task learning can consider different a set of embedding dimensions as tasks that have shared knowledge representation.
- Bayesian learning methods can infer parameters or weights of a neural network by marginalizing them out of the posterior distribution.

Bayesian inference

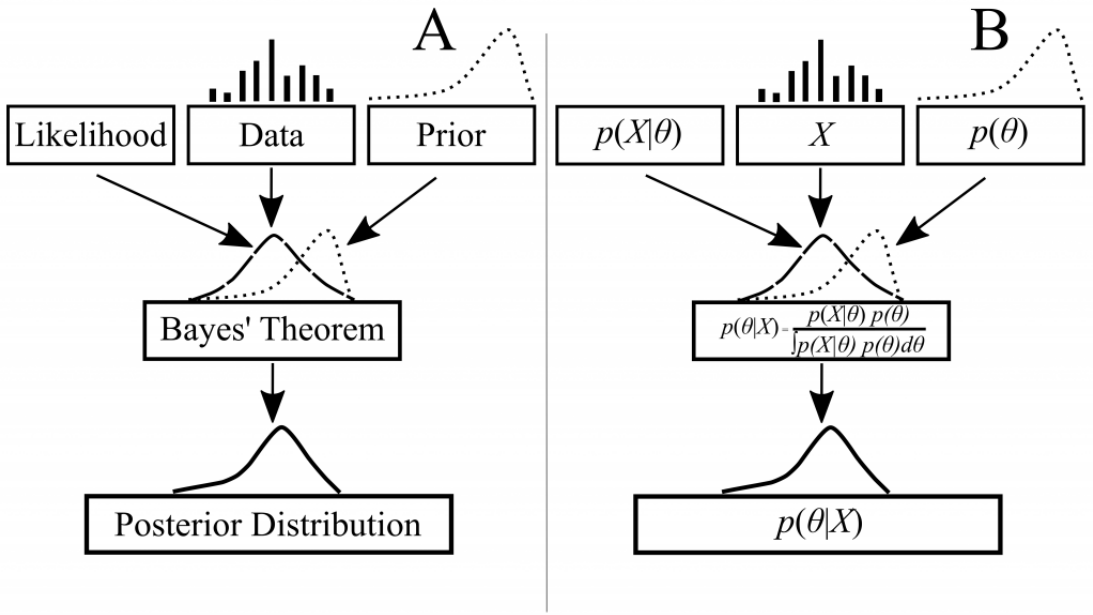


Figure: Bayesian inference overview

Markov Chain Monte Carlo sampling methods (MCMC) implement Bayesian inference that sample from a probability distribution. This is based on constructing a Markov chain after a number of steps that has the desired distribution as its equilibrium distribution.

Bayesian inference

The Theory: Bayesian inference

- Methodology of mathematical inference:
  - Choosing between several possible models
  - Extracting parameters for these models
- Bayes' Theorem:

$$p(w \mid D) = \frac{p(D \mid w)\,p(w)}{p(D)}$$

Likelihood — Prior Probability — Posterior Probability — Evidence

- Remove nuisance parameters by marginalisation
- Interesting ones remain

Rev Thomas Bayes 1702 - 1761

Figure: Bayesian inference overview

Bayesian inference

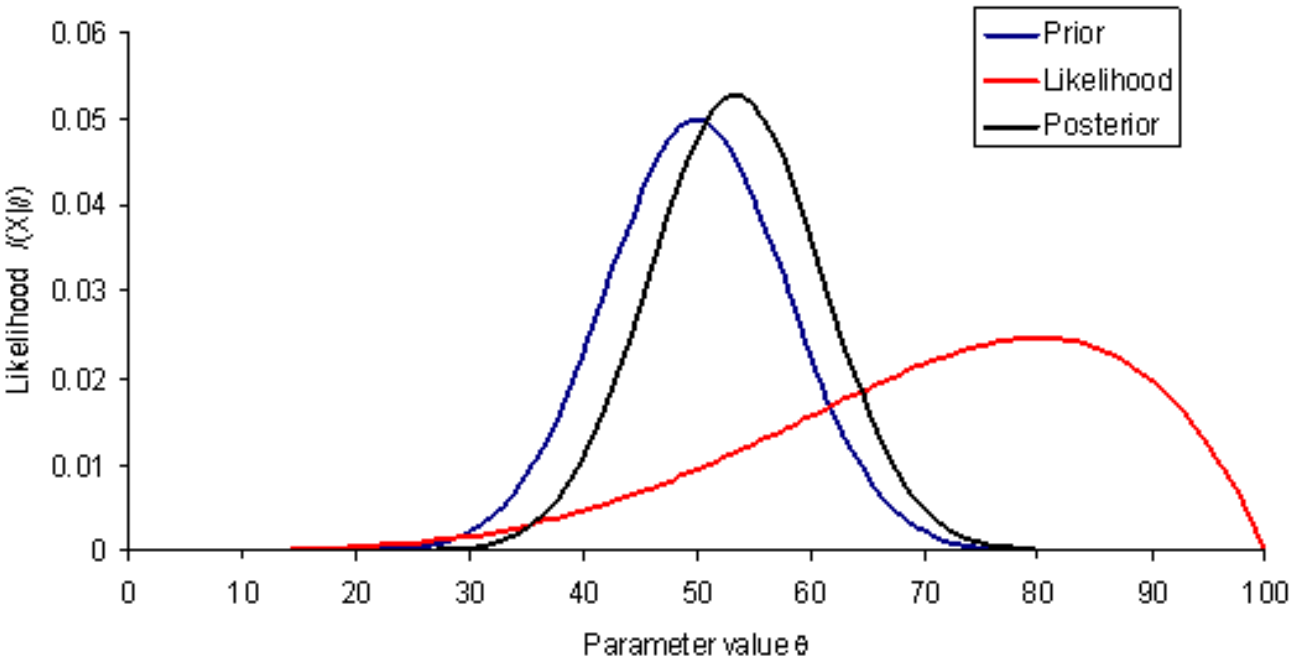


Figure: Bayesian inference overview

Background and Motivation

- Bayesian neural networks use Markov Chain Monti-Carlo (MCMC) methods for inference of network parameters such as the weights and biases.
- The motivations from related learning techniques is incorporated, in particular, multi-task learning for modular knowledge representation in neural networks.
- We present a Bayesian neural learning approach for dynamic time series prediction. The method provides uncertainty quantification through posterior distribution of weights and biases in a cascaded multi-task network.

The Robot Control Motivation

Consider a robot navigation problem that depends on input sensors that are controlled through a neural network.

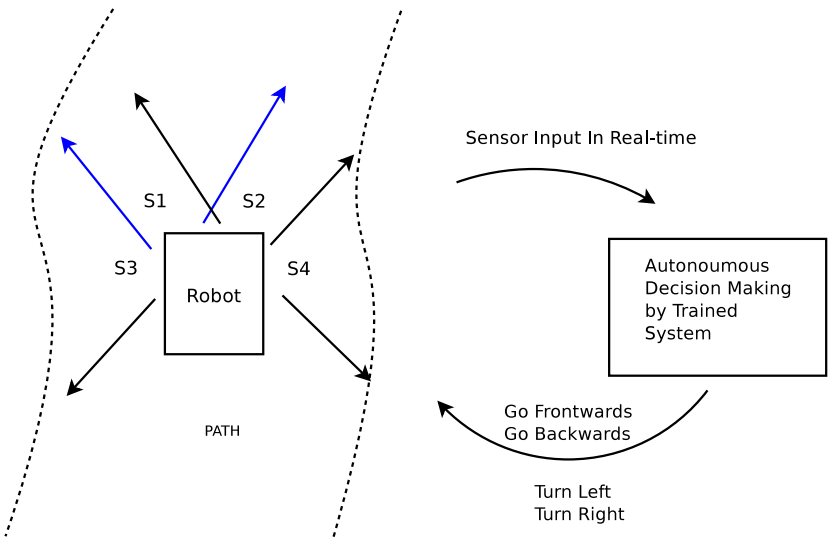


Figure: Control problem for autonomous robots based on sensor input (S1, S2, S3, and S4). The decision making can be done through a control system that features modular pattern classification.

Feature groups: Multi-task modular backpropagation

A feature group, X_m, is a subset of features. The overlapping subtasks $\Omega_1, \ldots, \Omega_m$ are defined as the union of selected feature groups $X_1, \ldots X_m$, for $m = 1, \ldots, M$, where M is the total number of feature groups, so that;

$$\begin{aligned} \Omega_1 &= [X_1] \\ \Omega_2 &= [X_1, X_2] \\ \Omega_3 &= [X_1, X_2, X_3] \\ \Omega_M &= [X_1, X_2, ..., X_M] \end{aligned} \tag{1}$$

Multi-task learning

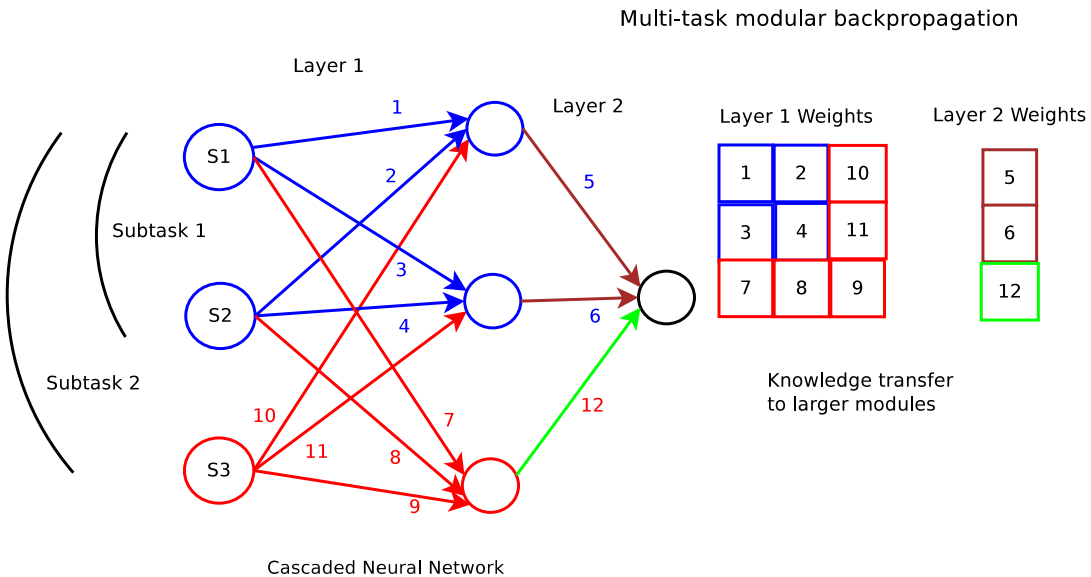


Figure: This cascaded modular neural network which can also be viewed and implemented as an ensemble of neural networks. The colours associated with the synapses in the network are linked to their encoding that are given as different modules. Subtask 1 employs a network topology with 2 hidden neurons while the rest of the modules add extra input and hidden neurons.

Method

Let $\hat{y}_{mt}$ denote a predicted value of y at time t based on a set of input features given in X_m

$$\begin{aligned} \hat{y}_{1,t} &= f(\theta_1; X_1) \\ \hat{y}_{2,t} &= f(\theta_2; X_2) \\ &\vdots \\ \hat{y}_{M,t} &= f(\theta_M; X_M) \end{aligned} \tag{2}$$

Method

Let $\hat{y}_{mt}$ denote a predicted value of y at time t based on a set of input features given in X_m

$$\begin{aligned} \hat{y}_{1,t} &= f(\theta_1; X_1) \\ \hat{y}_{2,t} &= f(\theta_2; X_2) \\ &\vdots \\ \hat{y}_{M,t} &= f(\theta_M; X_M) \end{aligned} \tag{3}$$

Method

The log likelihood for subtask m, $l_m(\boldsymbol{\theta_m})$, is

$$l_m(\boldsymbol{\theta_m}) = -\frac{P_M - 1}{2}\log(\tau^2) - \frac{1}{2\tau^2}\sum_{t \in \mathcal{A}_{\mathcal{D}_\mathcal{M},\mathcal{T}}} (y_t - \hat{y}_{m,t})^2 \qquad (4)$$

Algorithm

Data: Reconstructed univariate time series $\mathbf{y}$
Result: Posterior of weights and biases $p(\boldsymbol{\theta}|\mathbf{y})$
1: Reconstructed state-space vector for sub-task Ψ_s
2: Network for each sub-task $y_s = f(\theta_s, \Psi_s)$
4: Set parameters σ^2 ,ν_1, ν_2 for priors
Initialisation:
for *each sub-task* m **do**
  Propose initial weights and biases for the module: $\boldsymbol{\theta}_m^{[p]} = \mathcal{N}(0, \Sigma_\eta)$
end
Sampling:
for *each* k *until max-samples* **do**
  for *each sub-task* m **do**
    Step 1. Draw $\boldsymbol{\eta}$ from $\mathcal{N}(0, \Sigma_\eta)$
    Step 2. Propose set of weights and biases for the cascaded module given by the subtask: $\boldsymbol{\theta}_m^{[p]} = \Phi_m^{[k]} + \boldsymbol{\eta}$
    Step 3. Calculate posterior for subtask, Step 4. Draw from uniform distribution $u \sim \mathcal{U}[0, 1]$
    Step 5. Obtain acceptance probability α
    Step 6. Set $\Phi_m^{[k+1]} = \Phi_m^{[p]}$, if $u < \alpha$, otherwise $\Phi_m^{[k+1]} = \Phi_m^{[k]}$.
    Step 7. Transfer knowledge to subtask: $[\theta_m^{[k+1]} = [\theta_{m-1}, \Phi_m^{[k+1]}]$.
  end
end

Algorithm 1: Bayesian multi-task learning via MCMC sampler

Method

The loss for P_M predictions made using input features in X_m, can be calculated by root mean squared error.

$$rmse_m = \sqrt{\frac{1}{P_M} \sum_{t \in \mathcal{A}_{D_M,L}} (\hat{y}_{m,t} - y_t)^2} \tag{5}$$

Design of Experiments

- The Mackey-Glass, Lorenz, Henon and Rossler are the four simulated time series problems. The experiments use the chaotic time series with length of 1000 generated by the respective chaotic attractor.
- In all cases, the phase space of the original time series is reconstructed with the embedding dimensions for 3 datasets for the respective subtasks with embedding dimension $\Omega = s_1, s_2, s_3$ and time lag $T = 2$.
- We run two major experiments that are defined by the way the subtasks are created through different window size in the time series. In experiment 1, the subtasks are created by windows of size [2, 3, 4]. In experiment 2, the subtasks are created by windows of size [3, 5, 7].

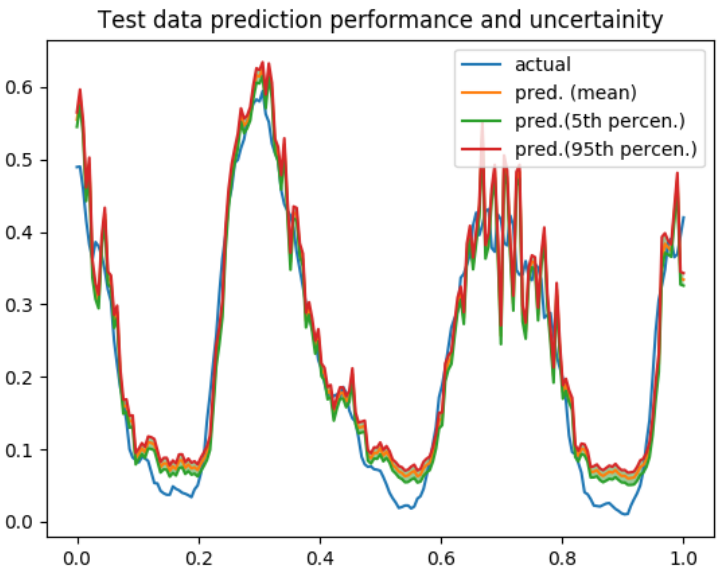
Results

Problem		Subtask	Train (mean)	Train (std)	Test (mean)	Test(std)	% Accept
Lazer	BMTL	1	0.0686	0.0098	0.0477	0.0043	16.40
		2	0.0621	0.0080	0.0490	0.0040	
		3	0.0647	0.0071	0.0453	0.0088	
	CMTL[1]	1	0.0644	0.0059	0.0766	0.0060	
		2	0.0666	0.0060	0.0875	0.0061	
		3	0.0926	0.0087	0.1164	0.0107	
Sunspot	BMTL	1	0.0501	0.0006	0.0495	0.0013	8.77
		2	0.0527	0.0081	0.0502	0.0017	
		3	0.0405	0.0001	0.0394	0.0001	
	CMTL[1]	1	0.0475	0.0068	0.04121	0.0067	
		2	0.0428	0.0044	0.0366	0.0040	
		3	0.0485	0.0062	0.0420	0.0050	
Mackey	BMTL	1	0.0296	0.0015	0.0297	0.0014	6.04
		2	0.0267	0.0016	0.0267	0.0016	
		3	0.0417	0.0001	0.0418	0.0001	
	CMTL[1]	1	0.0385	0.0050	0.0389	0.0047	
		2	0.0467	0.0048	0.0476	0.0051	
		3	0.0465	0.0047	0.0549	0.0061	

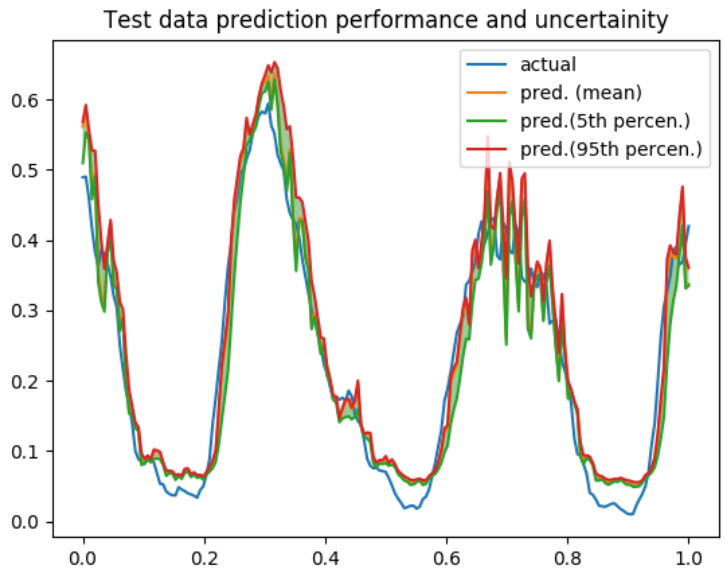
Results

Rossler	BMTL	1	0.0304	0.0023	0.0287	0.0021	6.17
		2	0.0292	0.0030	0.0276	0.0027	
		3	0.0320	0.0001	0.0305	0.0001	
	CMTL[1]	1	0.0454	0.0052	0.0475	0.0054	
		2	0.0448	0.0043	0.0464	0.0049	
		3	0.0498	0.0052	0.0528	0.0058	
Henon	BMTL	1	0.1743	0.0014	0.1756	0.0032	49.55
		2	0.1639	0.0029	0.1624	0.0027	
		3	0.1648	0.0009	0.1644	0.0011	
	CMTL[1]	1	0.0912	0.0137	0.0911	0.0131	
		2	0.1248	0.0126	0.1276	0.0127	
		3	0.1558	0.0121	0.1612	0.0121	
ACI-finance	BMTL	1	0.0447	0.0006	0.0234	0.0009	11.57
		2	0.0387	0.0003	0.0194	0.0002	
		3	0.0332	0.0002	0.0137	0.0003	
	CMTL[1]	1	0.0440	0.0060	0.0416	0.0106	
		2	0.0430	0.0041	0.0398	0.0075	
		3	0.0520	0.0050	0.0597	0.0080	

Results



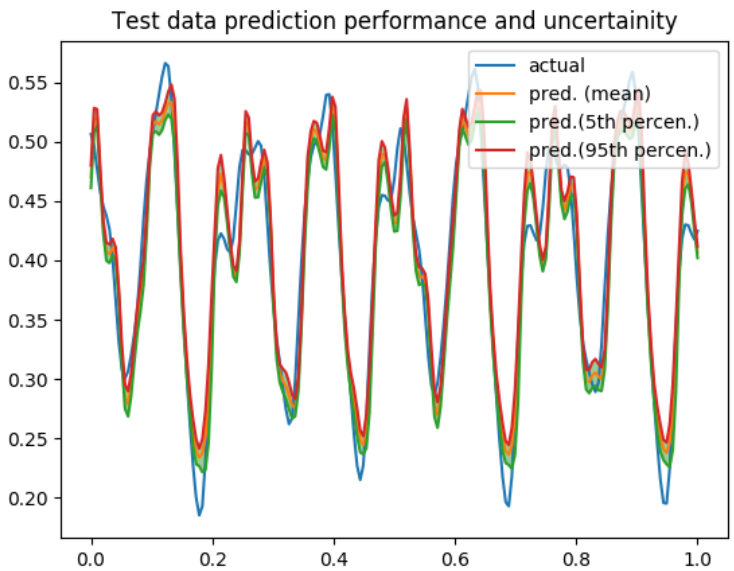
(a) Subtask 1



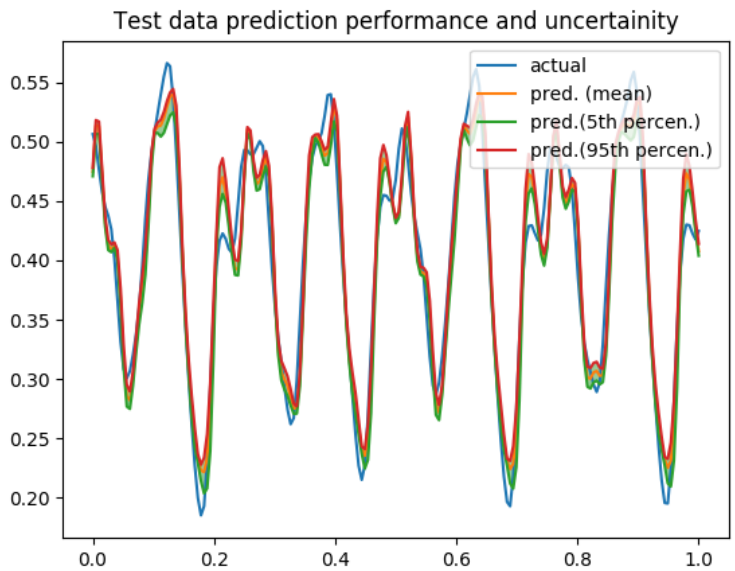
(b) Subtask 2

Figure: Sunspot performance evaluation across different subtasks (1, 2)

Results



(a) Subtask 1



(b) Subtask 2

Figure: Mackey Glass performance evaluation across different subtasks (1, 2)

Discussion

- We presented Bayesian multi-task learning for dynamic time series prediction via cascaded neural networks.
- The approach decomposed a single task learning problem of time series prediction into multi-task learning through subtasks that have inter-dependencies defined by the size of the window used for embedding.
- A major challenge was to adapt MCMC method for neural networks that featured subtasks that required transfer of knowledge.
- The training algorithm employed dynamic optimisation strategy where the knowledge from the foundation subtask was utilized in the subsequent subtasks through transfer learning.

Conclusions and Future Work

- The results show improvement for most of the problems when compared to related method from the literature.
- In future work, the performance of the proposed method can be further improved through Langevin dynamics where gradient decent is used to improve performance of random-walk MCMC methods.
- The approach can be used for pattern classification tasks and also for deep learning methods.

Thank You

More information - software included with paper :
https://rohitash-chandra.github.io