Learning to Predict Events On-line: A Semi-Markov Model for Reinforcement Learning

François Rivest, Richard Kohar, Najmatoullahi Amadou Boukary
Mathematics and Computer Science
Royal Military College of Canada

Introduction

Predicting upcoming events can be viewed as a time series forecasting problem.

But, training machine learning algorithms to predict the few next time steps is difficult and computationally expensive.

We propose to **predict the timing** of upcoming changes in observations (or states), **rather than the observations** at each time step, by minimizing the squared timing error.

In many reinforcement learning situations, it is sufficient to know the expected timing of various events or actions in order to behave optimally. For example, it may be sufficient for a robot to have an estimate of the remaining battery life and the time required to reach the recharge station.

We demonstrate how this algorithm can learn rapidly to predict music notes and how it can be used in conjunction with temporal-difference (TD) learning. The algorithm learns an approximated semi-Markov model allowing TD to learn a value function for a given policy in a partially observable environment.

Algorithm

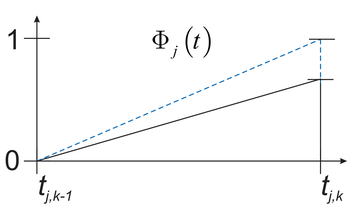
Given event input stream $x_i(t)$ and event stream $z_j(t)$.

Let weight $w_{j,i}$ and accumulator $\phi_{j,i}(t) = \phi_{j,i}(t-1) + w_{j,i}x_i(t)$.

We want $\Phi_j(t) = \sum_{i=1}^{N} \phi_{j,i}(t) = 1$ on events $z_j(t) = 1$ at $t = t_{j,k}$.

Observation time can be estimated using $a_{j,i}(t) = \phi_{j,i}(t) / w_{j,i}$.

Leading to the learning rule $\mathbf{w}_j \leftarrow \mathbf{w}_j - \alpha d_j \mathbf{a}_j(t_{j,k})$ where

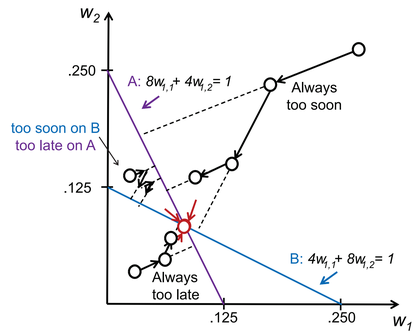


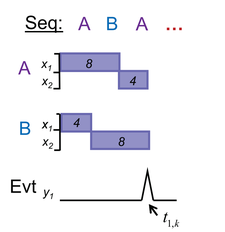
$$d_j = \frac{\mathbf{w}_j \mathbf{a}_j^T(t_{j,k}) - 1}{\mathbf{a}_j(t_{j,k}) \mathbf{a}_j^T(t_{j,k})}$$

Estimated remaining time:

$$\hat{y}_j(t) = \frac{1 - \phi_j(t)}{\mathbf{w}_j \cdot \mathbf{x}(t)}$$

Solution in Weight-space

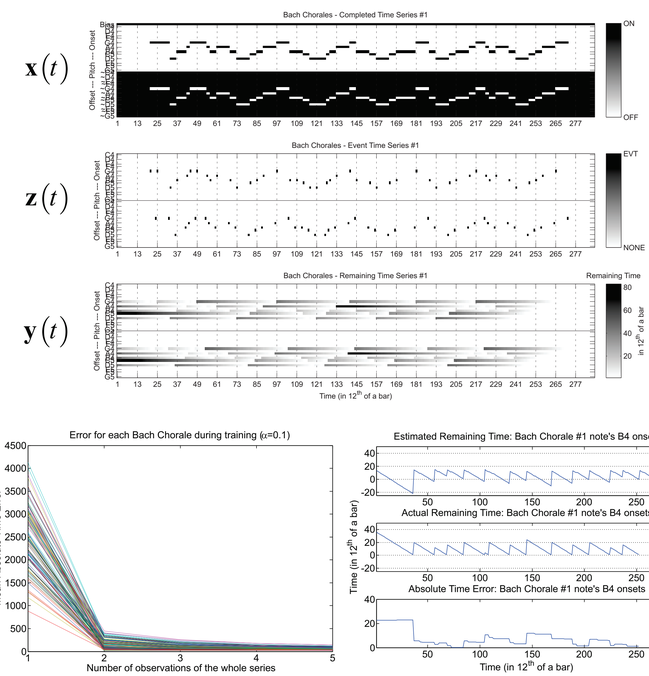




Target: $\Phi_1(t_{1,k}) = 1$

Learning rate: $\alpha = 0.5$

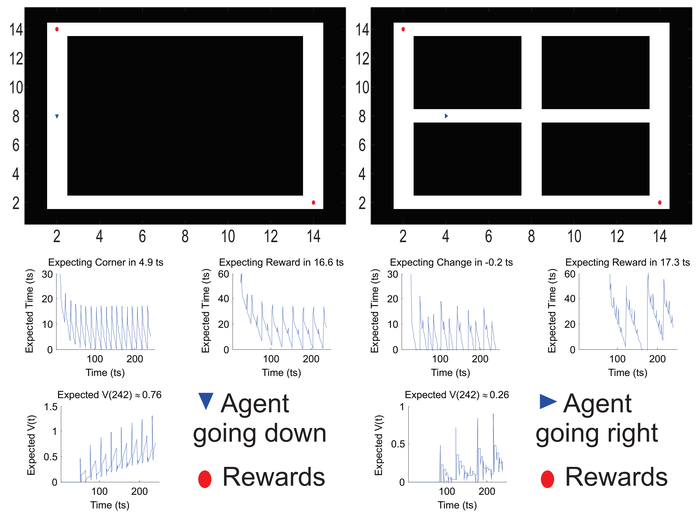
Predicting Notes' Onsets



Application to RL

TD: $V(s_t) \leftarrow V(s_t) + \alpha\left[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)\right]$

Using the remaining time in state: $V(t) = \gamma^{\max\left\{1, \min_j \{\hat{y}_j(t)\}\right\}} V(s_t)$



Conclusion

The new algorithm can learn rapidly to make timing predictions. These can then form an approximated semi-Markov model that TD can use to learn a value function.

References:
- Luzardo, Ludvig, Rivest (2013) Behavioural Processes.
- Rivest & Bengio (2011) ArXiv:1103.2382.

Contact:
- François Rivest: francois.rivest@{rmc.ca, mail.mcgill.ca}

Acknowledgments:
- ARP research grant from Royal Military College of Canada
- CDARP research grant from the Canadian Defence Academy