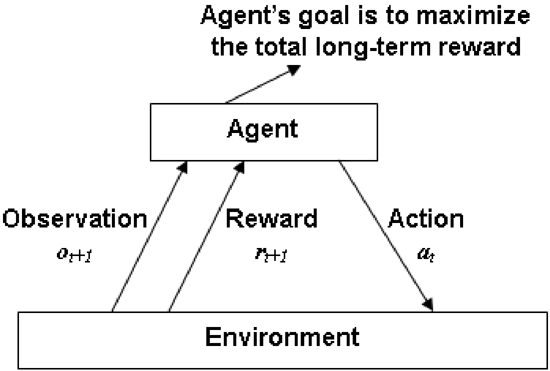
Context Tree Maximizing Reinforcement Learning (CTMRL)

March 2, 2012

Phuong Nguyen[1,2] (PhD student)
Joint work with **Peter Sunehag**[1] and **Marcus Hutter**[1,2]
Australian National University[1], *National ICT Australia*[2]

Reinforcement Learning (RL) Approach to Artificial Intelligence (AI)

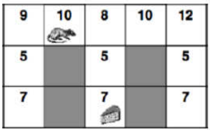
RL is an area of AI in which the agent learns a task through interactions with the environment



Problem formulation

$$
\begin{aligned}
h_t &= o_0 a_1 o_1 r_1 o_2 r_2 a_2 \dots o_t r_t \\
a_t &= \textbf{Agent}(h_t) \\
o_{t+1} r_{t+1} &= \textbf{Environment}(h_t a_t)
\end{aligned}
$$

- **General Reinforcement Learning (GRL) Problem**: find the agent function to maximize the total reward given that the environment's model and states are both unknown
  - **Example**:



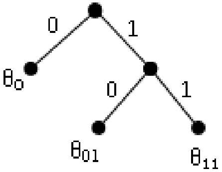
- **Special case**: Markov Decision Processes (MDP) where observations are states of the environment, $s_t = o_t$

Feature Reinforcement Learning - ΦMDP

$$
\begin{aligned}
h_t &= a_1 o_1 r_1 o_2 r_2 a_2 \ldots o_t r_t \\
a_t &= \textbf{Agent}(h_t) \\
o_{t+1} r_{t+1} &= \textbf{Environment}(h_t a_t) \\
s_t &= \Phi(h_t) \\
\textbf{Cost}(\Phi | h_n) &= \textbf{CL}(r_{1:n} | s_{1:n}, a_{1:n}) + \textbf{CL}(s_{1:n} | a_{1:n})
\end{aligned}
$$

Context tree maximizing for binary sequence prediction

- Context tree source



$\mathcal{S} = \{0, 01, 11\}$
$\theta_{01} = \mathbf{P}(\text{next\_bit} = 1|\text{current\_context} = s = 01)$

- Binary sequence prediction problem: find the optimal context tree given a history $x_{1:n} = 010010\ldots01$ and an initial context $x_{1-D:0} = 0100\ldots1$

F.M.J. Willems, Y.M. Shtarkov, and T.J. Tjalkens, *Context Tree Maximizing*, Conference on Information Sciences and Systems, Princeton University, 2000

Context tree maximizing for binary sequence prediction

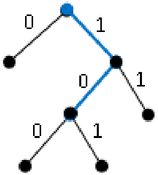
- Cost function

$$\min_{\mathcal{S} \subset \mathcal{C}_D} \left[\log \frac{1}{P_c(x_{1:n} | x_{1-D:0}, \mathcal{S})} + \Gamma_D(\mathcal{S}) \right]$$

$$P_c(x_{1:n} | x_{1-D:0}, \mathcal{S}) = \prod_{s \in \mathcal{S}} P_e(a_s, b_s)$$

  - Based on MDL (Minimum Description Length) principle, and arithmetic coding

Context tree maximizing for binary sequence prediction



- Iterative procedure to find the optimal tree
  - Maximized probability

$$
\begin{aligned}
P_{m,s}^{D} &:= 2^{-\Gamma_{D-d}(\mathcal{S}_{m,s}^{D})} \prod_{u \in \mathcal{S}_{m,s}^{D}} P_e^{a_{us}, b_{us}} \\
&= \max_{\mathcal{U} \subset \mathcal{C}_{D-d}} 2^{-\Gamma_{D-d}(\mathcal{U})} \prod_{u \in \mathcal{U}} P_e^{a_{us}, b_{us}}
\end{aligned}
$$

$$
P_{m,s}^{D} := \begin{cases} \frac{1}{2} \max(P_e(a_s, b_s), P_{m,0s}^{D} P_{m,1s}^{D}) & \text{for } 0 \le l(s) < D \\ P_e(a_s, b_s) & \text{for } l(s) = D \end{cases}
$$

Context tree maximizing for binary sequence prediction

- Iterative procedure to find the optimal tree
  - Maximizing set $S^D_{m,s}$

$$\mathcal{S}^D_{m,s} := \begin{cases} \mathcal{S}^D_{m,0s} \times 0 \cup \mathcal{S}^D_{m,1s} \times 1 & \text{if } P_e(a_s, b_s) < P^D_{m,0s} P^D_{m,1s}, \\ & \text{and } 0 \leq l(x) < D \\ \{\epsilon\} & \textit{else} \end{cases}$$

  - Theorem [*Willems et al, 2000*]: $\mathcal{S}^D_{m,\epsilon}$ is the optimal solution of the cost function for binary sequence prediction

Context Tree Maximizing Reinforcement Learning (CTMRL)

- Instance Context Tree: context tree with the instance set $\mathcal{X} = \{x^1, x^2, \ldots, x^{|\mathcal{X}|}\} = \{aor : a \in \mathcal{A}, o \in \mathcal{O}, r \in \mathcal{R}\}$ ($x_t = a_{t-1} o_t r_t$ is the instance at time t)
- Cost function

$$\begin{aligned} & \min_{S \subset \mathcal{C}_D} \left[\log \frac{1}{P_c(s_{1:n} r_{1:n} | a_{1:n}, h_0)} + \Gamma_D(\mathcal{S}) \right] \\ = & \min_{S \subset \mathcal{C}_D} \left[\Sigma_a \Sigma_s \log \frac{1}{P_e^{x|sa}} + \Gamma_D(\mathcal{S}) \right] \end{aligned}$$

where $\mathcal{S}$ is the state set of some instance context tree, $s_i = \Phi_{\mathcal{S}}(h_i), i = \overline{1, n}$; and $P_e^{x|sa}$ is the block probability of all instances

Context Tree Maximizing Reinforcement Learning

- Context tree maximizing iterative procedure
  - Maximizing probability

$$
\begin{aligned}
P^D_{m,s} &:= 2^{-\Gamma_{D-d}(\mathcal{S}^D_{m,s})} \prod_{a\in\mathcal{A}} \prod_{u\in\mathcal{S}^D_{m,s}} P_e^{x|usa} \\
&= \max_{\mathcal{U}\subset\mathcal{C}_{D-d}} 2^{-\Gamma_{D-d}(\mathcal{U})} \prod_{a\in\mathcal{A}} \prod_{u\in\mathcal{U}} P_e^{x|usa}
\end{aligned}
$$

$$
P^D_{m,s} := \frac{1}{2} \begin{cases} \max\left(\prod_{a\in\mathcal{A}} P_e^{x|sa}, \prod_i P^D_{m,x^i s}\right) & \text{if } 0 \le l(s) < D \\ \prod_{a\in\mathcal{A}} P_e^{x|sa} & \text{if } l(s) = D \end{cases}
$$

Context Tree Maximizing Reinforcement Learning

- Context tree maximizing iterative procedure
  - Maximizing state set

$$\mathcal{S}^D_{m,s} := \begin{cases} \bigcup_{x^i} \mathcal{S}^D_{m,x^i s} \times x^i & \text{if } \prod_{a\in\mathcal{A}} P_e^{x|sa} < \prod_{a\in\mathcal{A}} \prod_i P^D_{m,x^i s}, \\ & \text{and } 0 \le l(s) < D \\ \{\epsilon\} & \textit{else} \end{cases}$$

  - Theorem[direct extension]: $\mathcal{S}^D_{m,\epsilon}$ is the optimal solution of the CTM-GRL cost function
  - Problematic in estimating the multivariate block probability $P_e^{x|sa} := P_e^{x|sa}(n_{x^1}, n_{x^2}, \ldots, n_{x^{|\mathcal{I}|}})$

CTM-GRL: binarization and factorization

- The primary purpose of binarization is to overcome the estimation problem in $P_e^{x|sa}$
- Binarize observations, actions, rewards of a history. Each instance is represented in binary form as $x = aor = a[1 \dots l_a]o[1 \dots l_o]r[1 \dots l_r]$ $= ap = ap[1 \dots l_p]$. Consider the set of models $M = (M_1, \dots, M_{l_p}) \in \mathcal{C}_D \times \dots \times \mathcal{C}_{D+l_p-1}$

$$\begin{aligned} & \mathbf{Cost}(M|h_n, h_0) \\ = \ & \log \frac{1}{P_c(h_n|a_{0:n-1}, h_0, M)} + \sum_{i=1}^{p} \Gamma(M_i) \\ = \ & \sum_{i=1}^{p} \left[\sum_{t=1}^{n} \log \frac{1}{P_c(p_t[i]|h_t^i, h_0, M_i)} + \Gamma(M_i) \right] \end{aligned}$$

where $h_t^i = h_{t-1}a_{t-1}p_t[1 \dots i-1]$.

CTMRL algorithm

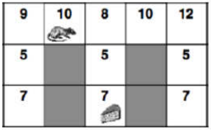
1. Generate a random history h
2. Learn (update) l_p binary CTMs based on history h (h' from the second iteration)
3. Join learnt contexts from each of the CTMs to form AOCT $\mathcal{T}$
4. Compute frequency estimates of state transition and reward probabilities of the MDP model $\widehat{M}$ based on states induced from tree $\mathcal{T}$ and history h
5. Use AVI to find an estimate of optimal action values $\widehat{Q}$ based on $\widehat{M}$
6. (Optional) Evaluate the current optimal policy induced from $\widehat{Q}$
7. $Q \leftarrow \widehat{Q} + \frac{R_{\max}}{1-\gamma}$ [[Optimistic Initialization]]
8. $h' \leftarrow$ Q-learning$(Q, S^{\mathcal{T}}, \mathcal{A}, Environment, n_i)$
9. $h \leftarrow [h, h']$
10. Repeat 2-9
11. $\widehat{Q}' \leftarrow$ Q-learning$(Q, S^{\mathcal{T}}, \mathcal{A}, Environment, n_q)$
12. $\pi^*(s) \leftarrow \text{argmax}_a \widehat{Q}'(s, a)$ for all $s \in \mathcal{S}^{\mathcal{T}}$

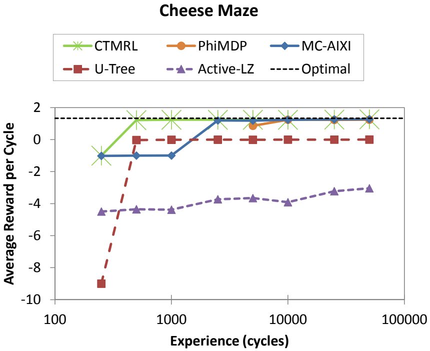
Related work

- ΦMDP
- MC-AIXI-CTW
- U-tree
- Active-LZ

Results - small domains

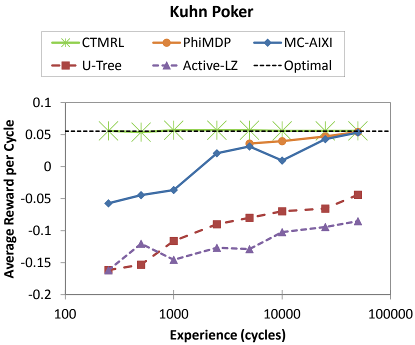
- Cheese maze





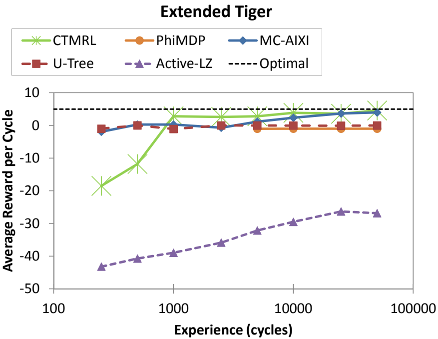
Results - small domains

- Kuhn poker



Results - small domains

- Extended tiger



- Other small domains: tiger, gridworld

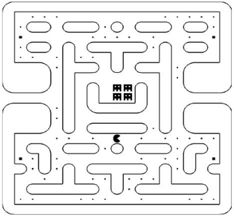
Results - large domain

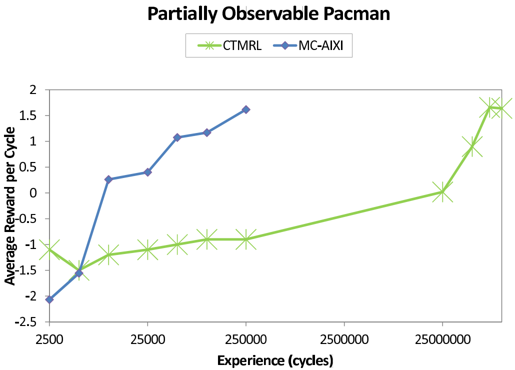
Modifications of CTMRL algorithm for large domains:

- Deletion of CTM trees after each learning loop (saving memory)
- Adding of unseen scenarios (dealing with huge observation space)
- Running Q-learning for a long time after the learning loop (solving MDPs with a large state space)

Results - large domain

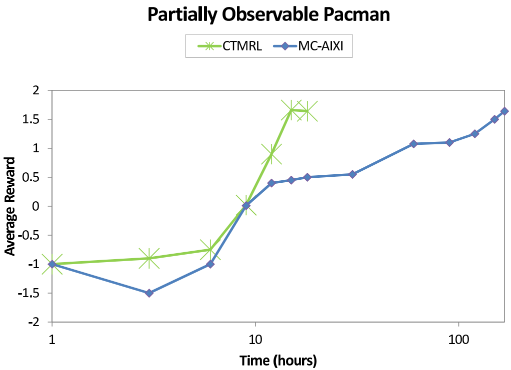
- Pacman





Results - large domain

- Pacman



Conclusion

- CTMRL is competitive with the state of the art MC-AIXI-CTW in terms of learning and superior to other competitors

- Compared to MC-AIXI-CTW, CTMRL is dramatically more efficient in both computation time and memory

Thank you!