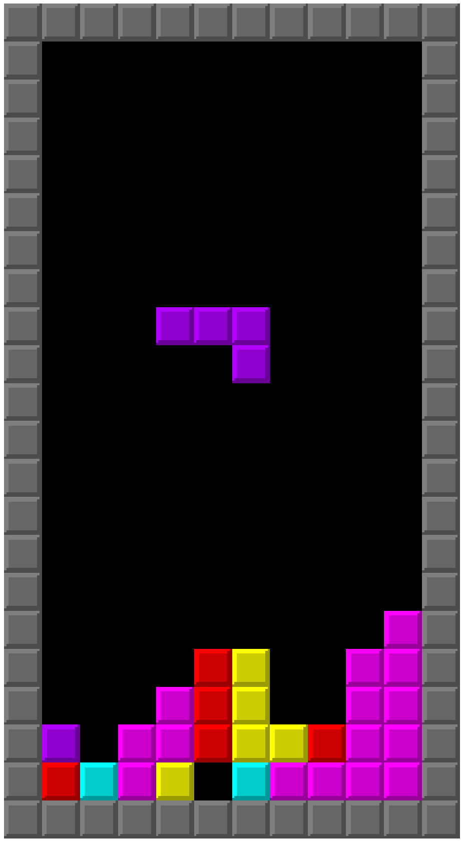
Approximate Linear Programming for Tetris

Ivan Sham



Outline

- Approximate Linear Programming
- Tetris
- Project Direction

Why use LP?

- Curse of dimensionality
  - Eg. Tetris has ~m*2^(w*h) states

- Approximate the cost-to-go function
  - Basis functions and weights

$$\tilde{J} = \sum_{k=1}^{K} r_k \phi_k \approx J^*$$

Approximate Linear Programming

Exact

$$\begin{array}{ll} \max & c^T J \\ \text{such that} & TJ \geq J \end{array}$$

Approximation

$$\begin{array}{ll} \max & c^T \Phi r \\ \text{such that} & T\Phi r \geq \Phi r \end{array}$$

$$TJ = \min_u (g_u + \alpha P_u J)$$

State Relevance Weights

Lemma: A vector r solves

$$\begin{array}{ll} \max & c^T \Phi r \\ \text{such that} & T\Phi r \geq \Phi r \end{array}$$

If and only if it solves

$$\begin{array}{ll} \min_r & \left\| J^* - \Phi r \right\|_{1,c} \\ \text{such that} & T\Phi r \geq \Phi r \end{array}$$

Error Bound for ALP

Loose $$\left\| J^* - \Phi \tilde{r} \right\|_{1,c} \leq \frac{2}{1-\alpha} \min_r \left\| J^* - \Phi r^* \right\|_\infty$$

Refined
$$\upsilon^T (J_{\tilde{\pi}} - J^*) \leq \frac{1}{1-\alpha} \left\| J - J^* \right\|_{1,c}$$
$$\upsilon(y) = \frac{1}{1-\alpha} (c(y) - \alpha \sum_x c(x) p_{\pi(x)}(x,y))$$
$$\tilde{\pi}(x) = \arg\max_{a \in A_x} \left(g(x,a) + \alpha \sum_{y \in S} p_a(x,y)(\Phi \tilde{r})(y) \right)$$

The Linear Programming Approach To Approximation Dynamic Programming

D.P. de Farias, B. van Roy, Operations Research 2003

Reduced Linear Programming

$$\begin{array}{ll} \max & c^T \Phi r \\ \text{such that} & g_a(x) + \alpha \sum_{y \in S} P_a(x,y)(\Phi r)(y) \geq (\Phi r)(x) \end{array}$$

All constraints $\forall x \in S, a \in A_x$

Reduced constraints $\forall (x,a) \in \chi$

Constraint Sampling Requirement

$$\Pr\{\left\|J^* - \Phi\hat{r}\right\|_{1,c} - \left\|J^* - \Phi\tilde{r}\right\|_{1,c} \leq \varepsilon\} \geq 1 - \delta$$

Feasibility of RLP

Let χ be the set of N constraints and

$$N \geq \frac{4}{\varepsilon}(K \ln(\frac{12}{\varepsilon}) + \ln(\frac{2}{\delta})) \qquad \varepsilon, \delta \in (0,1)$$

Then

$$\psi(V) \leq \varepsilon \quad \text{with probability} \quad 1-\delta$$

Where V is the set of constraints violated by optimal solution r

Error Bound for RLP

$$\left\| J^* - \Phi\hat{r} \right\|_{1,c} \leq \left\| J^* - \Phi\tilde{r} \right\|_{1,c} + \varepsilon \left\| J^* \right\|_{1,c}$$

On Constraint Sampling in the Linear Programming Approach to Approximate Dynamic Programming

D.P. de Farias, B. van Roy, Mathematics of Operations Research, August 2004

Tetris

- Intractable number of states ~m*2^(w*h)
- Objective:
  - Maximize rows clear before height reaches threshold
- States:
  - Board configuration
  - Shape of falling object
- Control:
  - Horizontal position
  - Rotation

RLP for Tetris

$$\begin{array}{ll} \max & \sum_{x \in \overline{\chi}} (\Phi r)(x) \\ \text{such that} & (T\Phi r)(x) \geq (\Phi r)(x) \end{array}$$

$$\forall x \in \overline{\chi}$$

$$(TJ)(x) = \min_{a \in A_x} \{g(x,a) + \alpha (P_a J)(x)\}$$

Basis Functions

- Height of each column (10)
- Height difference successive columns (9)
- Maximum height (1)
- Number of holes (1)
- Static value of 1 (1)

- Total of 22 basis functions

Sample of States

- Only sample constraint every M time steps
  - Play N games
  - Keep track of states every for every M-th block

Bootstrapping

- Start with policy u_0

- Generate sample of states X_k with u_k
- Solve RLP to get u_{k+1}
- Repeat

Demo

- Only implemented the approximation architecture
- No DP or LP (weights are not updated)

What’s the plan?

- Implement RLP solution
- Add a few other basis functions

- Implement other approaches
  – Temporal Difference (Bertsekas 1996)
  – Others if I have time

- Anything else that come up while I’m coding