Chunking with Reinforcement Learning

Joseph Xu

University of Michigan

Soar Workshop 28

Outline

- *Chunking with uncertain knowledge*
- Calculating uncertainty in Reinforcement Learning
- Evaluation
- Discussion / Nuggets & Coals

New Knowledge Types

- New modules in Soar introduce new types of knowledge
  - Q-values in RL
  - Episodic memories
  - Semantic memories/Clusters
- Used to make decisions even when suboptimal/incorrect/changing
  - RL makes decisions with randomly initialized Q-values
  - Episodic memories retrieved with the same cue change with experience

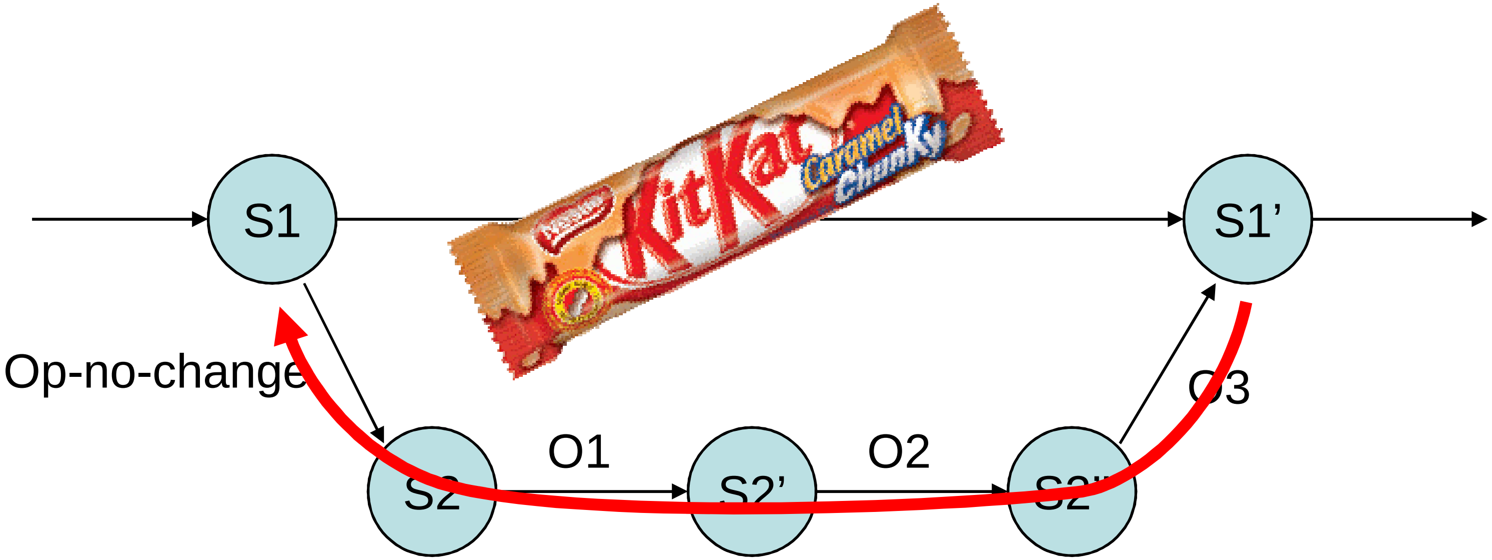
Chunking

- Summarizes problem solving in subgoals
- Assumes decisions are correct instead of “best we can do for now”
- Chunking over new types of knowledge will result in permanent suboptimal rules
- Solution:
  - Hold off chunking decision processes until we’re confident that they are correct or nearly optimal

Determining When to Chunk

- Need general mechanism to inform chunking algorithm about confidence in decisions
- Associate probabilities with operator selection
  - P(operator O should be selected in state S | Knowledge)
  - Design separate mechanisms to calculate probabilities for decisions conditioned on each type of new knowledge
  - Decisions made with only symbolic preferences have probability 1

Chunking over Probabilities



P(O1 should be selected in S2 | K) = 0.95
P(O2 should be selected in S2' | K) = 0.95
P(O3 should be selected in S2'' | K) = 0.9

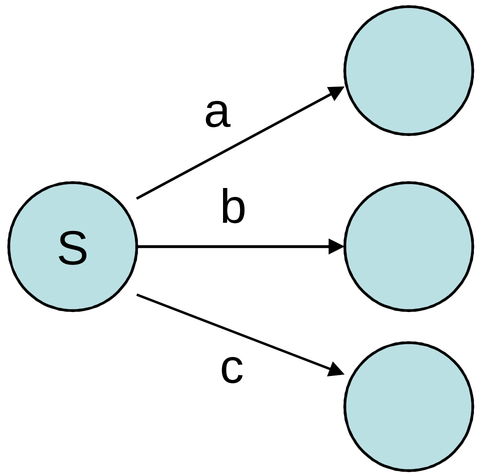
P(S1' should follow S1 | K) = Probability that we should chunk
= 0.95 * 0.95 * 0.9 = 0.81225 > 0.8

Outline

- Chunking with uncertain knowledge
- *Calculating uncertainty in Reinforcement Learning*
- Evaluation
- Discussion / Nuggets & Coals

Probabilities in RL

- Operator with highest **true** Q-value is the correct operator to select



$$\text{Should select a iff } Q(s,a) > Q(s,b) \wedge Q(s,a) > Q(s,c)$$

$$P(\text{should select a}) = P\left(\hat{Q}(s,a) > \hat{Q}(s,b)\right) \times P\left(\hat{Q}(s,a) > \hat{Q}(s,c)\right)$$

Finding P(Q(s,a) > Q(s,b))

- Establish **confidence bounds**
  - "97% confident that true Q is in [Q^{min}, Q^{max}]"

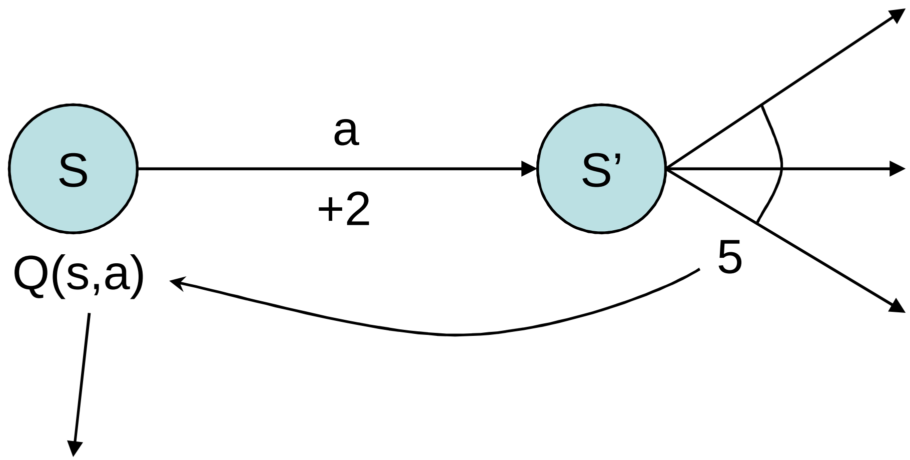
$$P(Q_1 > Q_2) = \begin{cases} P\left(Q_1 \in \left[Q_1^{\min}, Q_1^{\max}\right]\right) \cdot P\left(Q_2 \in \left[Q_2^{\min}, Q_2^{\max}\right]\right) = .97^2 & \text{if } Q_1^{\min} > Q_2^{\max} \\ 0 & \textit{otherwise} \end{cases}$$

Interval Estimation

- Kaebling 1993
- Estimate true Q with median of last N sampled Q values
- For each (s,a), maintain sliding window of N most recent Q values

Interval Estimation

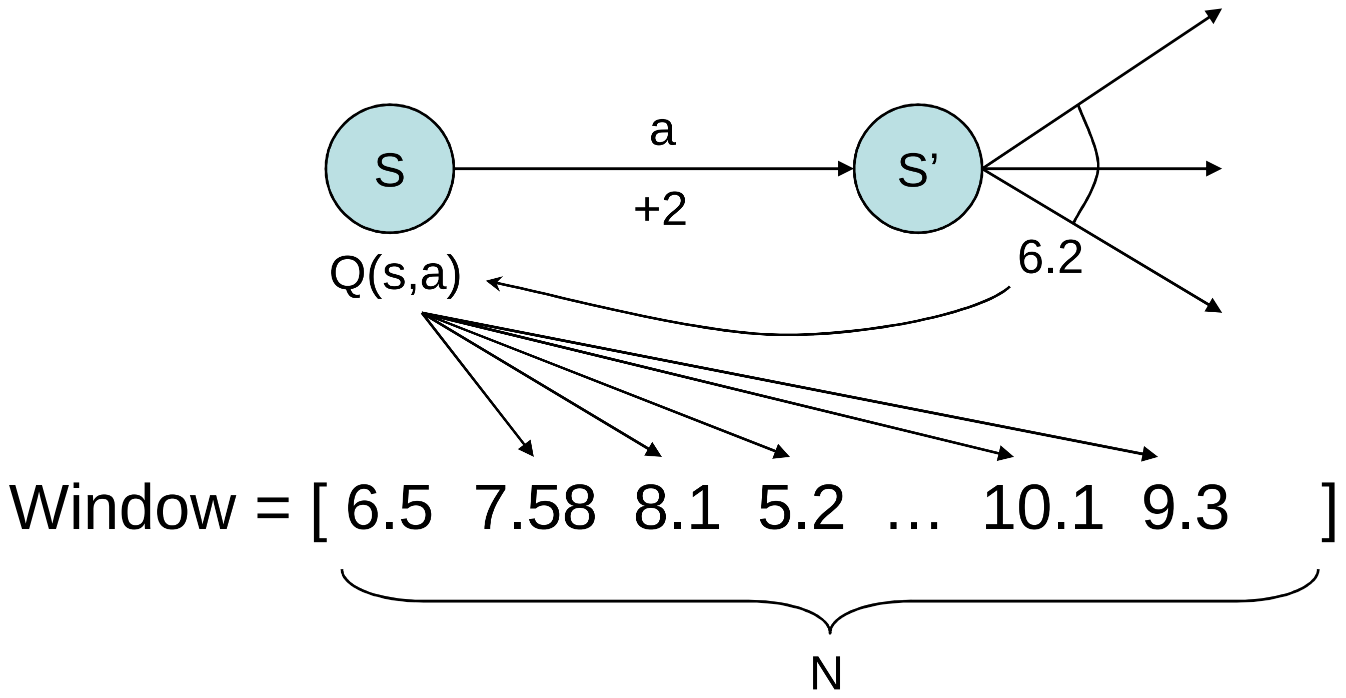
- Estimate true Q with median of last N sampled Q values (Kaebling 1993)



Window = [6.5]

Interval Estimation

- Estimate true Q with median of last N sampled Q values (Kaebling 1993)



Interval Estimation

Window = [5.2 6.5 7.58 8.1 … 9.3 10.1]

1 2 3 4 n-1 n

r Q_r Q_s s

$$P(\text{exactly } k \text{ samples } < Q_{med}) = P(Q_k < Q_{med} < Q_{k+1}) = 0.5^n \cdot \binom{n}{k}$$

$$P(Q_r < Q_{med} < Q_s) = \sum_{k=r}^{s-1} 0.5^n \cdot \binom{n}{k} \geq 0.97$$

Choose r and s that are as close as possible but still meet constraints

Interval Estimation in Soar

- Keep track of confidence bounds for all state-operator pairs
  - If the sample window hasn't filled up, assume that $Q(s,a) \in [-\infty, +\infty]$
  - Otherwise calculate as described
- If confidence bounds are separated, chunk over decision
- Can only chunk when all windows are filled

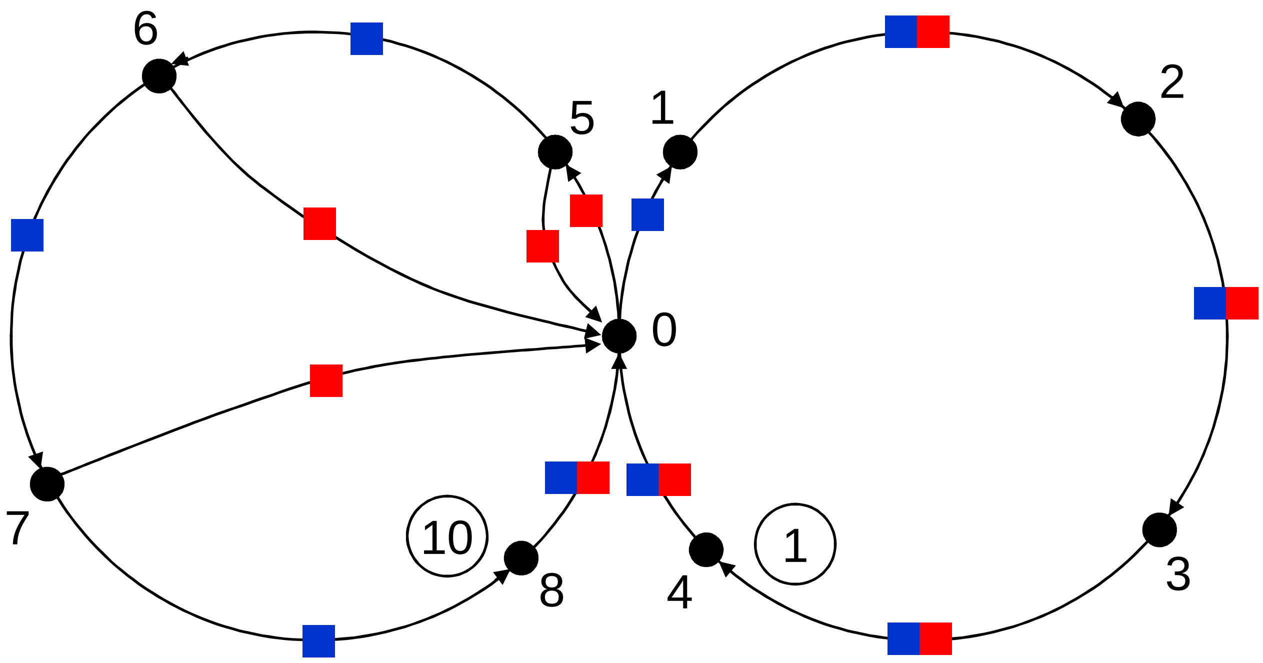
Outline

- Chunking with uncertain knowledge
- Calculating uncertainty in Reinforcement Learning
- *Evaluation*
- Discussion / Nuggets & Coals

Evaluation Criteria

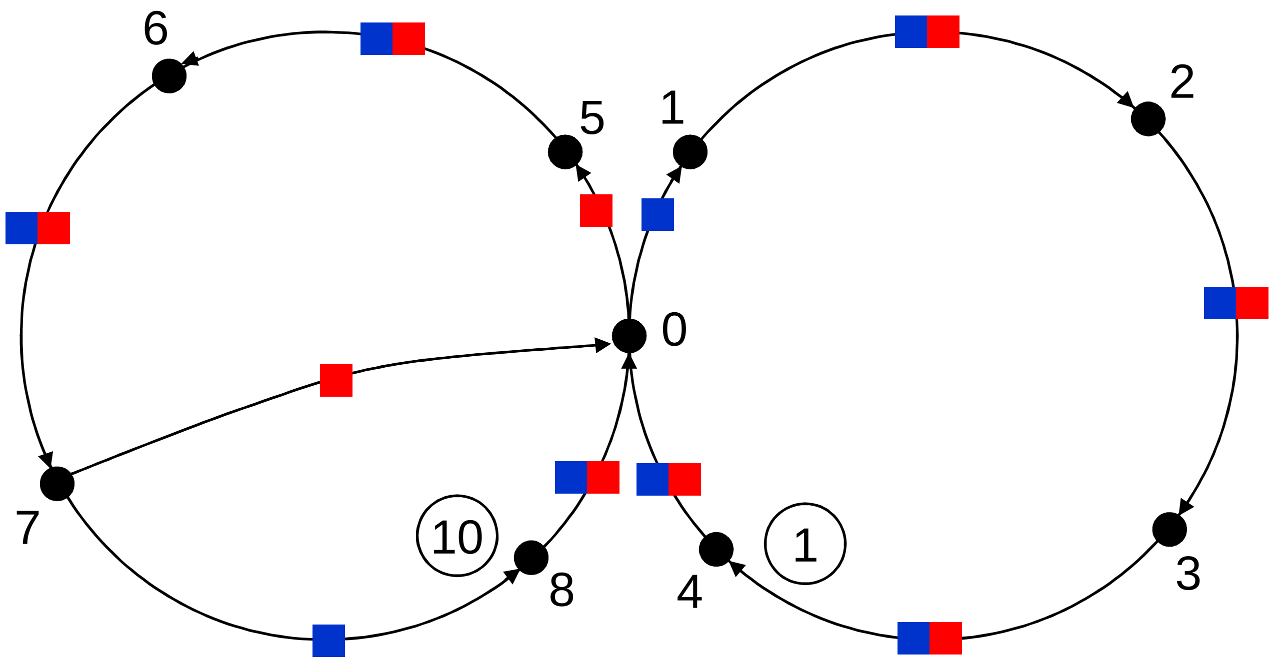
- Do intervals overlap to prevent chunking when RL policy still nonstationary?
- Can interval estimation be tricked into separating intervals by complex environments?

Evaluation Environment

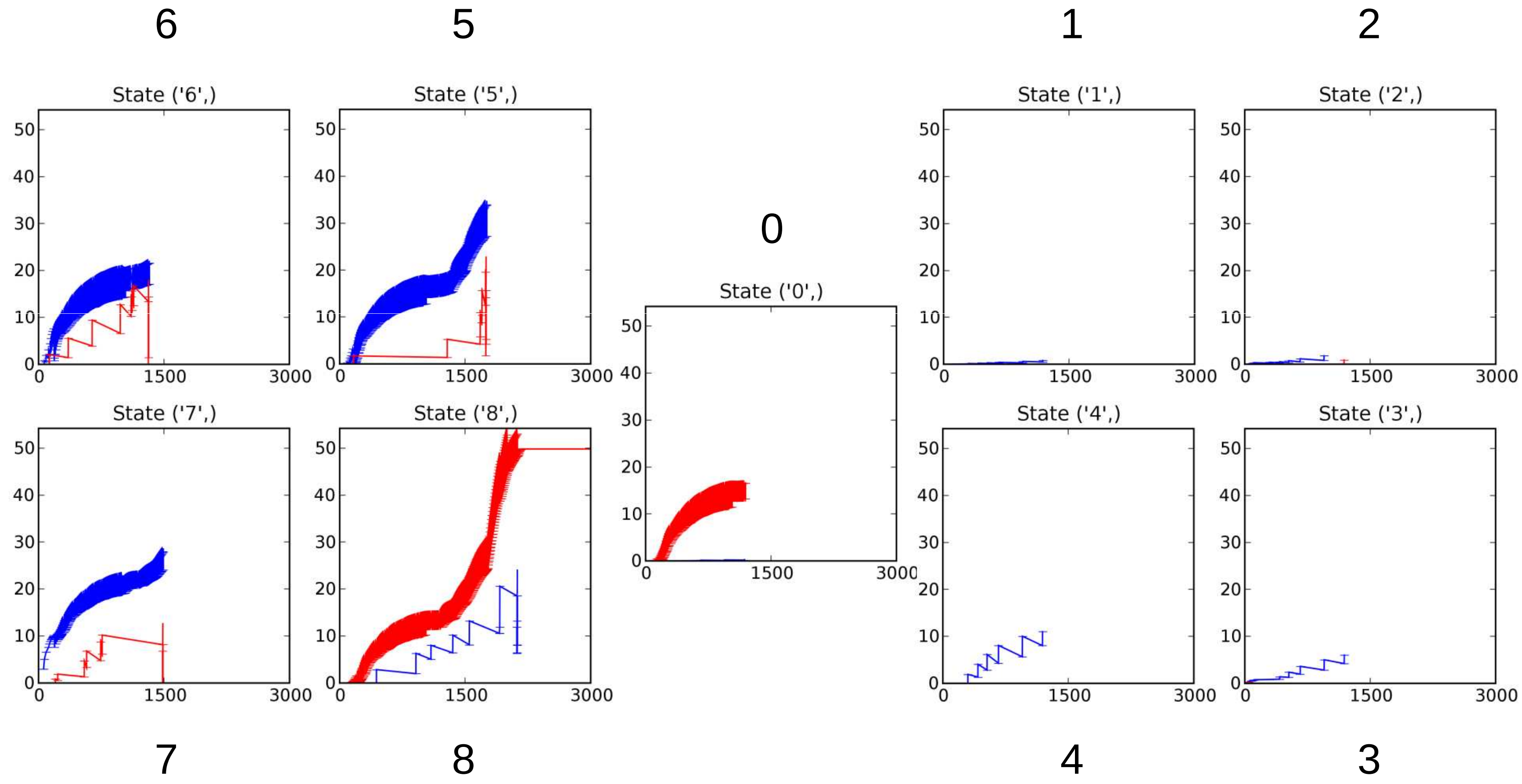


Adapted from Kaebling 1993

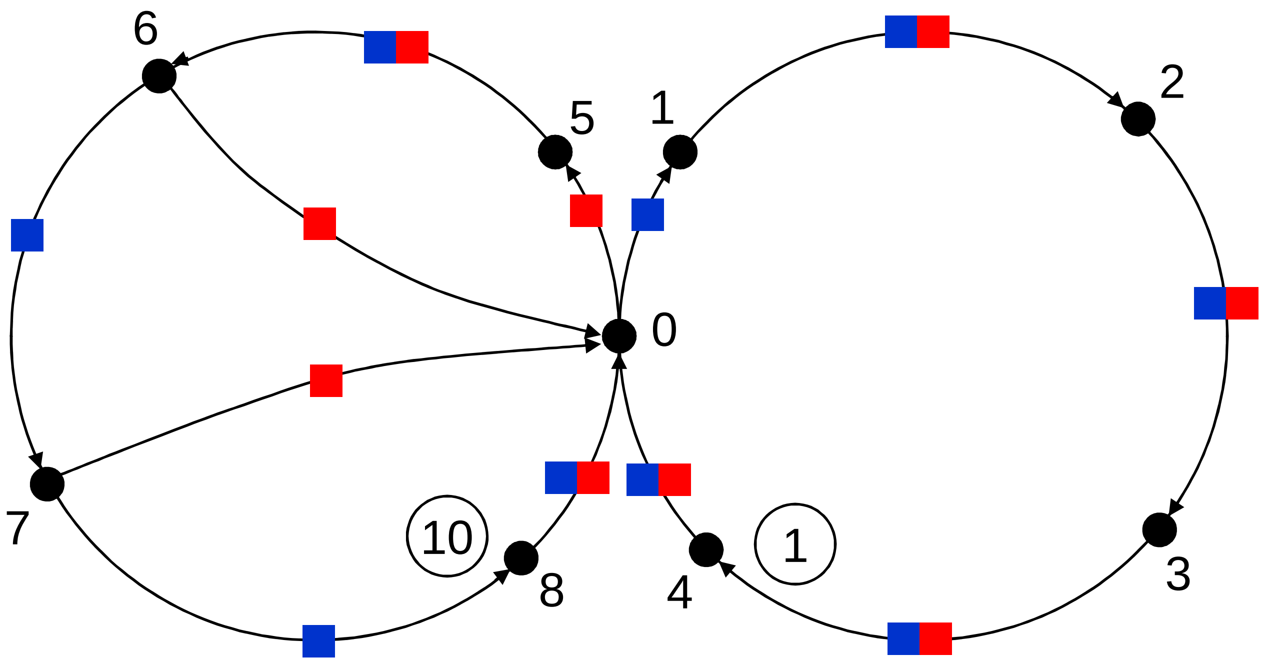
Easy Environment



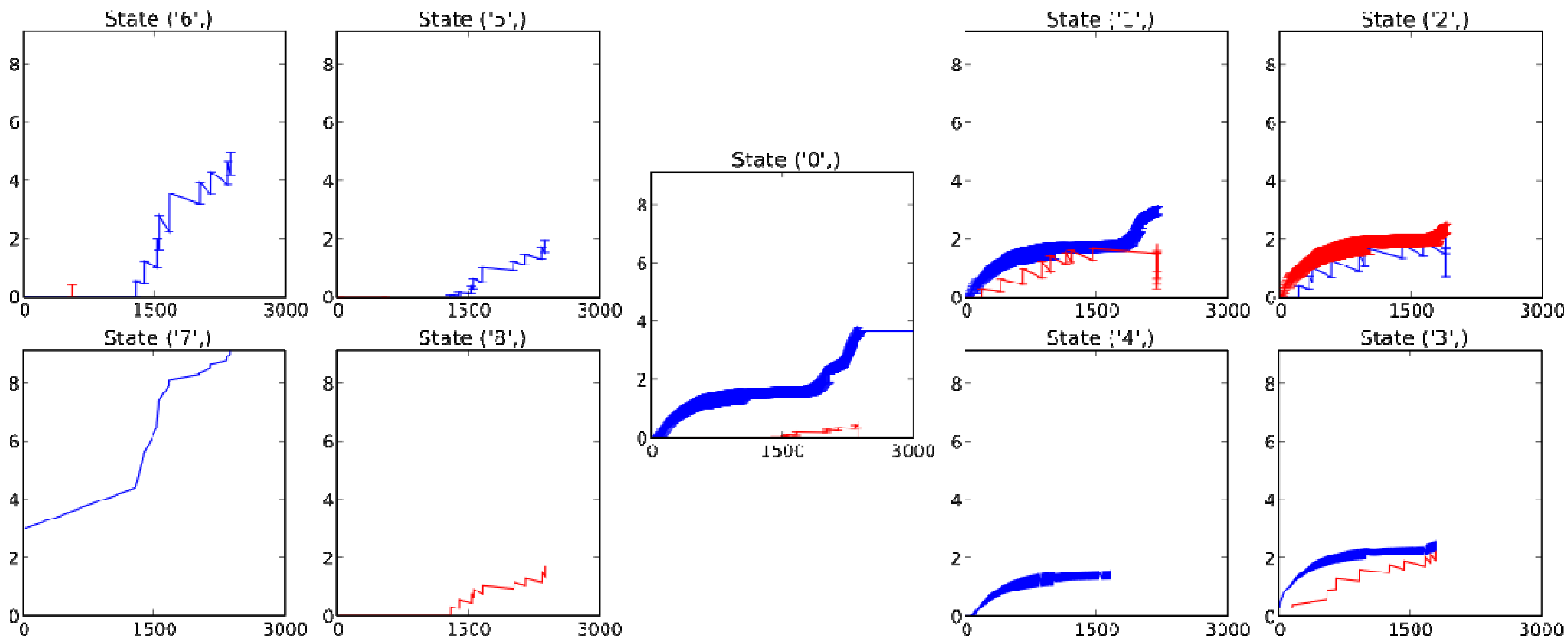
N=20, r=5, s=16 (P=0.97)



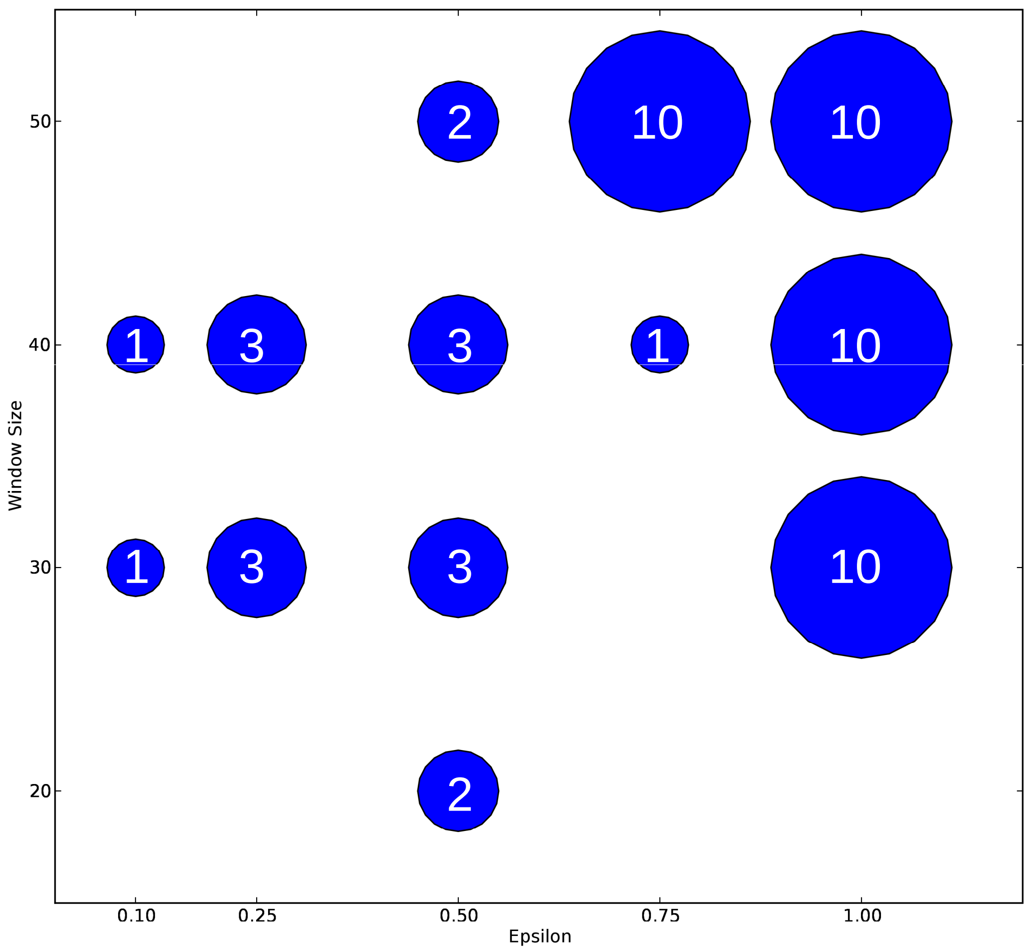
Harder Environment



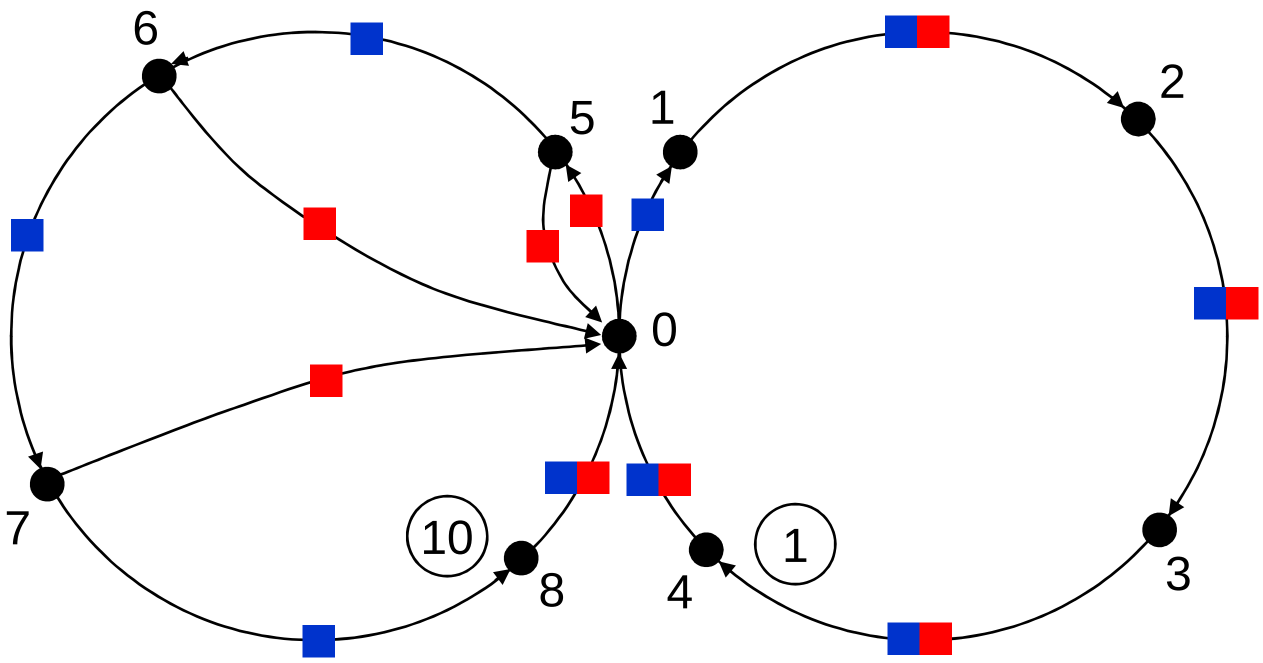
N=20, r=5, s=16 (P=0.97)



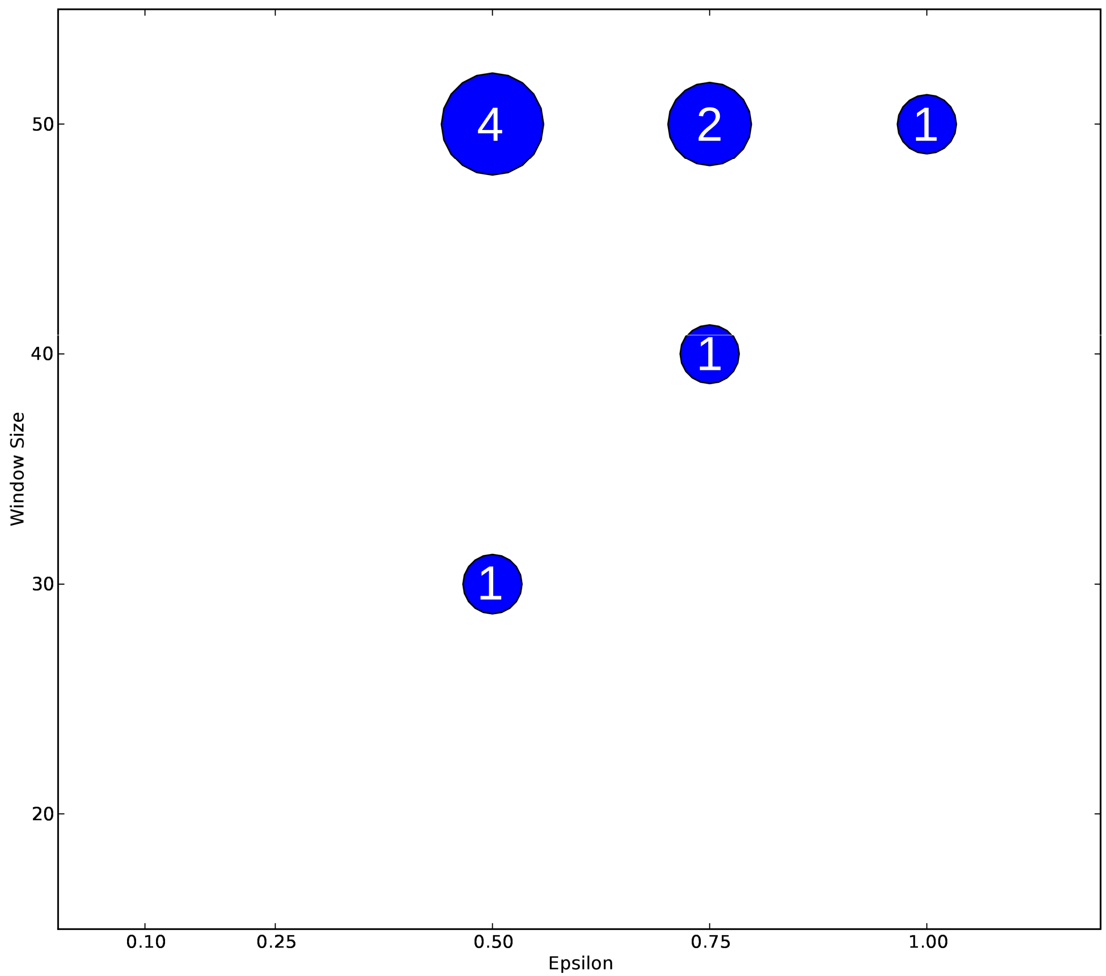
Window Size vs. Exploration



Hardest Environment



Window Size vs. Exploration



Conclusion

- Interval estimation doesn't provide theoretical guarantees of boundedness
- Empirically, it can overcome some trickiness in the environment
- Window size and exploration factor have to be tweaked on a per-environment basis

Conclusion

- Nuggets
  - Interval estimation works empirically
  - Provides some protection against chunking bad decisions
  - Computationally very cheap
- Coals
  - No theoretical guarantees on bounds
  - More parameters to adjust
- Two other methods being investigated:
  - Hoeffding inequality
  - Bayesian estimation