Spectral Learning for Structured Partially Observable Environments

Lucas Langer

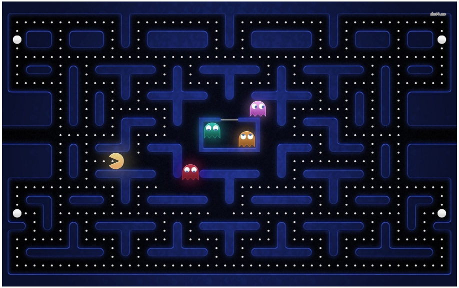
Supervisors: Borja Balle, Doina Precup

August 29, 2015

Partially Observable Environments

Structured Environments

1. Goal: Predict observations
2. Plan: Exploit patterns
3. Example: Pacman



PSR: The Timing Case

- Model: Predictive State Representation (PSR)
  For timing we have one observation symbol: σ
  Notation: σ: one time unit, σ^k: k time units
- PSR defined by: $\langle \alpha_0, \{A_\sigma\}, \alpha_\infty \rangle$
  α_0: Initial weighting on states $1xn$
  A_σ: Transition matrix nxn
  α_∞: Normalizer $nx1$
- PSRs compute probabilities of observations
  $f(\sigma^k) = \alpha_0 \cdot A_\sigma^k \cdot \alpha_\infty$
  Ex: HMMs

Spectral Learning of PSRs

Step 1: Represent Data as a Hankel Matrix

Step 2: Singular Value Decomposition

Step 3: Pick Model Size

Step 4: Learn PSR: $\langle \alpha_0, \{A_\sigma\}, \alpha_\infty \rangle$

The Base System

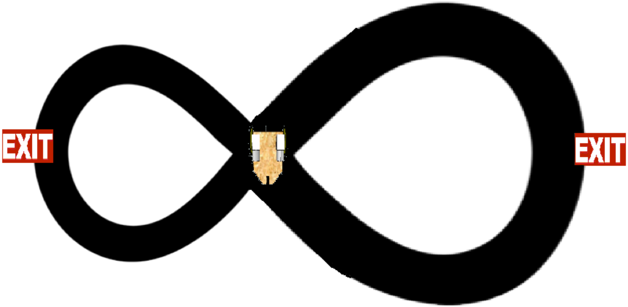
- Idea: Learn $\{A_\sigma, A_{\sigma^2}, A_{\sigma^4}, A_{\sigma^8}, ...A_{\sigma^N}\}$ as extra transition operators
- Timing queries: $f(\sigma^{11}) = \alpha_0 \cdot A_{\sigma^8} \cdot A_{\sigma^2} \cdot A_{\sigma^1} \cdot \alpha_\infty$
- Motivation:

  1) Express transitions directly

  2) Faster queries

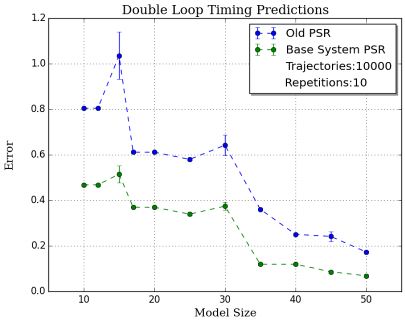
Timing with the Base

Agent drives around loops until leaving through an exit state.

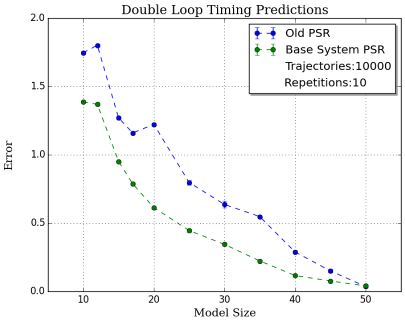


Base System Performance for Loops

64-16 Loop Lengths



47-27 Loop Lengths

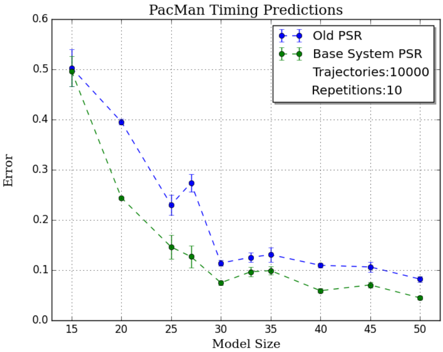


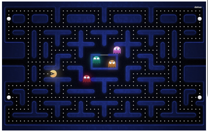
- Base System dominates for smaller models

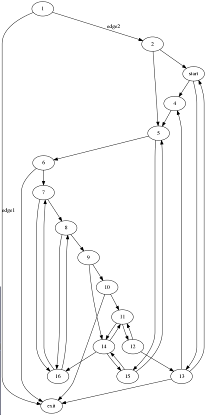
$$||f - \hat{f}|| = \sqrt{\sum_{x \in observations} (f(x) - f(\hat{x}))^2}$$

Pacman Labyrinth

Timing Predictions



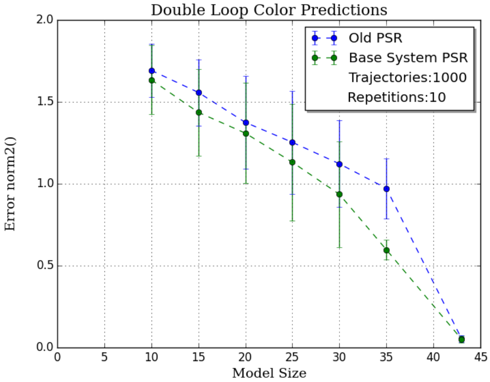


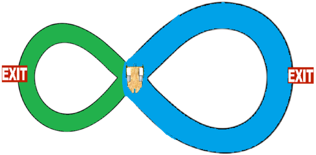


$$||f - \hat{f}|| = \sqrt{\sum_{x \in observations} (f(x) - f(\hat{x}))^2}$$

Wall Color Predictions

We paint the first loop green and the second loop blue





$$||f - \hat{f}|| = \sqrt{\sum_{x \in observations} (f(x) - \hat{f(x)})^2}$$

Picking the Base System

- Observations: $\{"a^{30}":10, "a^{60}":5, "b^{18}":15\}$
  Desired Base System: $A_{a^{30}}$, $A_{b^{18}}$, A_a, A_b

- Solution: iterative greedy heuristic
- Substring properties: **long**, **frequent**, **diverse**

- Entropy view of structure

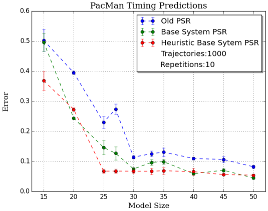
Computing with the Base System

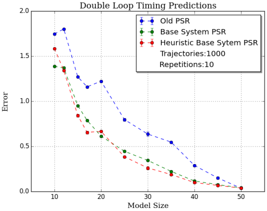
- How should we execute queries?
  Goal: **minimize number of matrices**

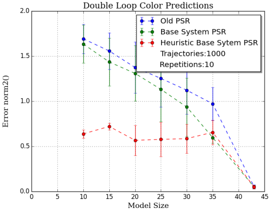
  Query string: "abcacb", Base System $= \{A_{ab}, A_{bca}, A_{cb}, A_a, A_b\}$
  Desired partition: "a—bca—cb"
  Computation: $f(abcacb) = \alpha_0 \cdot A_a \cdot A_{bca} \cdot A_{cb} \cdot \alpha_\infty$

- Solution: dynamic programming

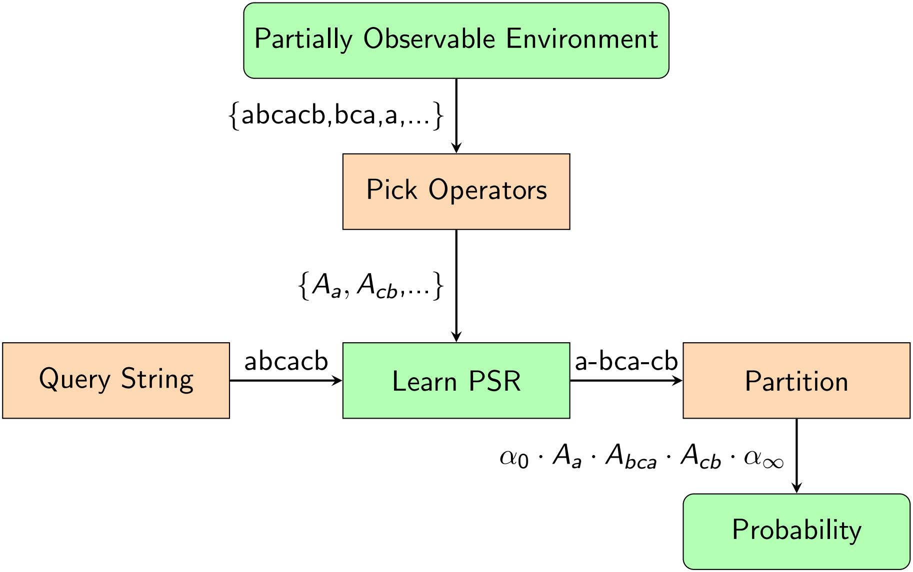
Performance of Heuristics







The Big Picture



Questions?