Fast Feature Selection for Reinforcement-Learning-based Spoken Dialog Management: A Case Study

Lihong Li\textsuperscript{1}  Jason D. Williams\textsuperscript{2}  Suhrid Balakrishnan\textsuperscript{2}

\texttt{lihong@cs.rutgers.edu  jdw@research.att.com  suhrid@research.att.com}

\textsuperscript{1}Department of Computer Science, Rutgers University, Piscataway, NJ 08854
\textsuperscript{2}AT&T Labs – Research, 180 Park Avenue, Building 103, Florham Park, NJ 07932

**Example Application: Dialer**

- AT&T internal voice dialer system w/ 50K employees in directory
- Phone types: cell, and/or office
- Can be modeled as a (partially observable) Markov decision process
  - Available actions
    - Reward function
      - AskName: “First and last name?”
      - -1 per action
      - ConfirmPhoneType: “Cell phone – is that right?”
      - +20 for correct transfer
      - -20 for incorrect transfer

**Problem Statement**

- LSPI is very data-efficient, but time-expensive
- Complexity: $O(k^3)$ – $k$ is number of features
- Fast feature selection is needed
- Existing techniques do not scale well

**Methodology**

1. Run temporal difference to get a coarse estimate of $\{w_1, \ldots, w_k\}$
2. Fast (but rough) policy evaluation w/ Temporal Difference

**Feature Selection Procedure**

1. Evaluate $\pi$ w/ Least-Squares Temporal Difference
2. Select most significant features

**Results**

- Comparisons
  - HC-Baseline
    - Running in production for the past 3 yrs
    - Hand-crafted policy
  - RL-Baseline
    - Choose representative states
    - Build an approximate model
    - Run value iteration to get $\pi$
- Number of raw features: 3456
- Number of features selected: 400
- Averaged over 10 runs
- Use 1000 dialogs for evaluation

**Challenges**

- Observations are noisy (due to speech recognition error)
  - Need handle partial observability
- Can maintain distributions over hidden variables (e.g., user intentions)
  - Extract features from conversation history
- Problem space is large
  - Need value function approximation
  - Linear function approximation w/ Least-Squares Policy Iteration (LSPI)