Better Generalization with Forecasts

Tom Schaul
Mark Ring
Which Representations?

Type: Feature-based representations
(state = feature vector)

Quality 1: Usefulness for linear policies

Quality 2: Generalization

Representations?
- FOL + policy
- flat
- black box generative
- DBN
- hierarchical?
- RBF, linear
- supervised/unsupervised (e.g., DBN)
- basis functions
- who cares?
- can we learn it?/efficient
- expressiveness & expressiveness & structure exposure?
- amenability to exploration
- robustness & regularizable
- planning suitability
- controllability
- biocomputing
Outline

✓ Motivation

• Representation
  • Predictive State Representations
  • General Value Functions, aka “Forecasts”
  • Simplified subclass of Forecasts

• Evaluating Generalization

• Results
• Question/test: “Will I hit the wall if I take a step left and then a step back?”
• Expected answer = feature \( \phi \)
• Defined as a set of testable predictions
  • Observable quantities (wall sensor)
  • Conditional on action sequence (step left, step back)
  • Open-loop (ends at \( t+2 \))
• $\phi = "After how many steps will I encounter a door if I head to the wall in front of me, and follow it clockwise?"

• General Value Functions
  • More general questions
  • Closed-loop: arbitrary length sequences

• We call them "Forecasts"
Forecast Components

- Conditional on an option: following a policy (straight + clockwise) until termination (door)
  - $I$: states of interest
  - $\beta(s)$: termination probability
  - $\pi$: policy

- Target value (expectation): any function of the state, cumulative
  - $c(s)$: accumulated value before termination
  - $z(s)$: final value upon termination in $s$
Simplified Forecasts

• Constant components:
  • \( c(s) = 0 \)
  • \( I = \text{all states} \)

• Defined by only a target set of states \( T \)
  • \( z(s): 1 \text{ if } s \in T, \ 0 \text{ elsewhere} \)
  • \( \beta(s): 1 \text{ if } s \in T, \ 1 - \gamma \text{ elsewhere} \)
  • \( \pi: \text{implicitly defined: maximizes the expected } z \)

→ Only one free parameter: \( T \)
→ Output: feature vector \( \phi(s) \)
Outline

✓ Motivation
✓ Representations

• Evaluating Generalization
  • Ideal vs. estimated forecast values
  • Canonical forecast ordering
  • Quality measures

• Results
Forecast Values

• Distinguish:
  • Forecast definition (“question”)
    • From target set $T$
  • Ideal forecast value (true “answer”)
  • Forecast value estimate (approximation)
    • can be learned

• Focus: quality of representation
  • Use ideal forecast values as features
  • We can ignore the learning issues (i.e., we can cheat!)
  • Namely: policy iteration with transition model
Forecast Generation (1)

• Canonical (breadth-first) exhaustive generation
  • First layer based on observations
  • Forecasts can build upon other forecasts
  • Unique ordering (lexicographic tie-breaking)
Forecast Generation (2)

1. Initially: observations define target sets $\mathcal{T}$
2. Compute ideal forecast values $\phi$ from $\mathcal{T}$
   - Cheat 1: transition model (infinite experience)
   - Cheat 2: knowledge of state
3. Threshold $\phi$ for new candidate $\mathcal{T}$ sets
4. (Ignore redundant sets)
5. Go to step 2, until no $\mathcal{T}$ is left
Evaluation of Feature Sets

• Quality measures:
  1. Optimal policy using linear function approximation (LFA) on features (LSTD+PI)
  2. Distance between estimated external value $V'$ (using LFA on features) and true $V^*$ (MSE)

• Generalization:
  • consider random subset of states (e.g. 50%)
  • train the LFA based on this limited experience
  • see how that $V'$ generalizes to the remaining states
Outline

✓ Motivation
✓ Representations
✓ Evaluating Generalization

• Results
  • Two mazes
  • Simplistic agent
    • 1 binary observation (wall sensor)
    • 1 binary action (forward/rotate left)
  • Comparison to PSRs as baseline
4-Arm-Maze: PSRs

![Graph showing performance and MSE with varying number of features.]
4-Arm-Maze: Forecasts
7-Rooms-Maze: PSRs

Better Generalization with Forecasts
7-Room-Maze: Forecasts
Forecast Features
Summary

Forecasts as representations:

- Temporal abstraction (closed-loop)
- Simple structure
- Useful features for linear evaluation and control
- Good generalization with linear FA

Future work

1. Removing the cheats: using learned estimates instead of ideal forecast values
2. Smarter forecast generation than exhaustive