Kaius Perttilä

Active Gesture Recognition using POMDP

T-61.6020

Reinforcement Learning—Theory and Applications

Outline

An example of a POMDP application

- problem statement
- application
- results

- paper by Trevor Darrell and Alex Pentland, MIT Media Lab

Gesture recognition problems

- dynamic environment
  - closed laboratory room vs. walking street
- available picture quality
  - optics, resolution, lighting, angles, amount of cameras
- processing power
  - real-time(ish) applications & high resolution

Solution in practice

- low-resolution general view
  - gives a world picture used to control the active cameras
  - used to identify the user’s body parts
- movable foveated views
  - able to track a limited area
  - would consist of several cameras in practice
  - actively focused at relevant body parts, e.g. face

Application assumptions

- active cameras focused on user's hands and face
  - unsufficient number of active cameras to track all body parts
  - which body part to track unknown *a priori*
- method requirements:
  - perceptual selection that learns from experience
  - apapts to only partial observations of world

Application assumptions

- routines to track body movements using a low-resolution carmera exist

- static low-resolution camera inadequate to model the world
  - body parts contain hidden states
  - otherwise there would be no problem

Active Gesture Recognition

Model consists of:

1. variables describing the state of the world
2. portions of the world are only revealed by a moving fovea
3. the fovea can be actively controlled
4. an accept command

AGR

- Modelled as an POMDP
  - states
  - obseravtions
  - actions
  - reward function
- Practical limitations limit the explicit solving of the transitions
  - hidden state learning (Q-learning)

AGR states 1/2

- Person tracker:
  - *person-present* { true, false }
  - *left-arm-extended* { true, false }
  - *right-arm-extended* { true, false }
- Foveated gesture recognition:
  - *face* { neutral, smile, suprise }
  - *left-hand* { neutral, point, open }
  - *right-hand* { neutral, point, open }

AGR states 2/2

- Internal state of vision system:
  - head-foveated { true, false }
  - left-hand-foveated { true, false }
  - right-hand-foveated { true, false }

AGR actions

- Four foveation commands:
  - *look-body*
  - *look-head*
  - *look-left-hand*
  - *look-right-hand*

- *Special command:*
  - *accept*

AGR algorithm

- Modified Q-leaning
  - store (a[t], r[t], o[t]) tuples
  - store an individual q[t]–value for each tuple
  - for each time instance
    - rate each tuple, and select K most similar instances of them
    - calculate Q-value (mean [K most similar])
    - select action with highest Q-value (or random)
    - update the q-values for each tuple in the history

Multiple objects 1/2

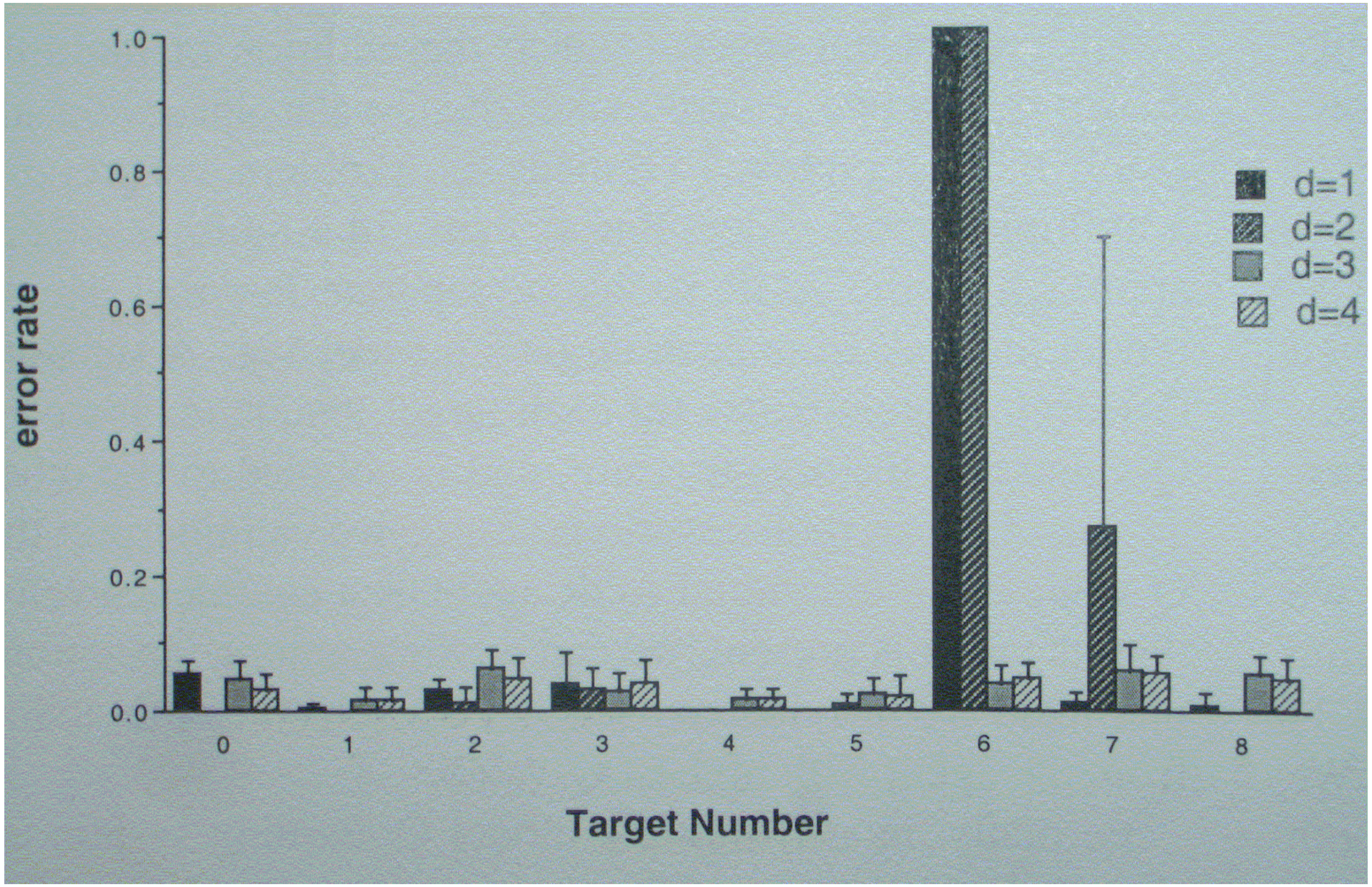
- presented system handles focusing action descision when there is one target
- several targets (=*accept* states) leads to a complex value space and slow convergance

Multiple objects 2/2

- Solution:
  - each target has its own learning agent, own history and Q-values
  - observations shared between agents
  - select action based on the highest Q-value among the agents

Results 1/2

- AGR was connected to person tracking and gesture analysis systems



Results 2/2

- complex patterns require extensive learning
- Limitations:
  - environment unknown
  - one active camera used
    - real applications need several point of views
  - person tracking and gesture analysis systems not presented

Questions?