Project 3 Recitation

Friday, October 26, 2018 2:57 PM

Q1: Exact Inference Observe Method

- NoisyDistance: if noise, there is a ghost
- Emission Model: P(B|A)
- Loop on all the legalPositions:
  - True distance -> B
  - Position (State) -> A
  - P(A|B) = P(B|A) * P(A)
  - allPossible[State] = emissionModel[TD] * self.beliefs
  - P(A) -> self.beliefs
  - Normalizes allpossible -> self.beliefs
  - No NoisyDistance -> Ghost is in jail
    - Set the probability of the jail position to 1

Q2: Exact Inference Elapse Time Method

- Loop on all positions -> called pos
  - newPosDist = self.getPositionDistribution() -> Distribution
  - Loop through newPosDist -> newpos, prob
    - allPossible[newPos] += self.beliefs[pos] * prob
- Normalize allPossible

Q3: Buster Agents chooseAction

- Multiple Ghosts
- LivingGhostPositionDistributions-> list of Counter objects

- Closest square, mindist
- Loop on LGPD
    - Find the max probability per distribution
    - Take the state (max prob state)
        - Self.distancer.getDistance(maxprobstate, pacman)
        - If distance < mindist
            - Closestsqure = maxprobstate
- Loop on all legal (legal actions)
    - Distance = getDistance(closestSquare, Actions.getSuccessor(pacman, legalact))
- Return the best action

Q4: Particle Filtering Observe

- Initialize Uniformly
    - Pigeonhole the number of particles
- Observe method
    - NoisyDistance is none -> probs[jailPos] = 1
    - Loop on the particleList
        - Td, put that into the emission model
        - Probs[particle] += emissionmodle[td]
    - If (probs.totalCount() ==0)
        - initializeUniformly
    - Else
        - Normalize
        - newpartlist
        - For x in range(self.numparticles)
            - Newpartlist.append(util.sample(probs))

- Self.particleList = newpartlist

Q5: Particle Filtering with elapse time

- Newlist =[]
- Loop on self.particlelist
  - newPosDist -> provided in the comments
  - Newlist.append(util.sample(newposdist))
- Set particlelist to newlist

GetBeliefDistribution (Q4)

Give all the particles 1 per time they appear and normalize

EC

Particle Filter from before and modify it to work with several ghosts

Key points

- List of noisy distances
- Particles are NOT a single position
- Say you have 4 ghosts
  - Particle looks like
  - ((1,2),(3,2),(4,1),(5,2)) -> some probability
- Initialize Uniformly
  - List(itertools(self.legalPostions, repeat = self.numghosts))
  - Random.shuffle(list from above)
  - Make the first x your particles
- If your ghost is in jail

- getParticleWithGhostInJail(particle, 3)
- ((1,2),(3,2),(4,1),(5,2)), 3 ->
  ((1,2),(3,2),(Jail,Jail),(5,2))