

Homework 4

AutoGrader

Starting on 4-25 at 16:54:12

Question q1

=====

```
*** PASS: test_cases\q1\1-small-board.test
*** PASS: test_cases\q1\2-long-bottom.test
*** PASS: test_cases\q1\3-wide-inverted.test
```

Question q1: 2/2

Question q2

=====

```
*** PASS: test_cases\q2\1-product-rule.test
***      Executed FactorEqualityTest
*** PASS: test_cases\q2\2-product-rule-extended.test
***      Executed FactorEqualityTest
*** PASS: test_cases\q2\3-disjoint-right.test
***      Executed FactorEqualityTest
*** PASS: test_cases\q2\4-common-right.test
***      Executed FactorEqualityTest
*** PASS: test_cases\q2\5-grade-join.test
***      Executed FactorEqualityTest
*** PASS: test_cases\q2\6-product-rule-nonsingleton-var.test
***      Executed FactorEqualityTest
```

Question q2: 3/3

Question q3

=====

```
*** PASS: test_cases\q3\1-simple-eliminate.test
***      Executed FactorEqualityTest
*** PASS: test_cases\q3\2-simple-eliminate-extended.test
***      Executed FactorEqualityTest
*** PASS: test_cases\q3\3-eliminate-conditioned.test
***      Executed FactorEqualityTest
*** PASS: test_cases\q3\4-grade-eliminate.test
***      Executed FactorEqualityTest
*** PASS: test_cases\q3\5-simple-eliminate-nonsingleton-var.test
***      Executed FactorEqualityTest
*** PASS: test_cases\q3\6-simple-eliminate-int.test
***      Executed FactorEqualityTest
```

Question q3: 2/2

Question q4

=====

```
*** PASS: test_cases\q4\1-disconnected-eliminate.test
***      Executed FactorEqualityTest
*** PASS: test_cases\q4\2-independent-eliminate.test
```

```

***      Executed FactorEqualityTest
*** PASS: test_cases\q4\3-independent-eliminate-extended.test
***      Executed FactorEqualityTest
*** PASS: test_cases\q4\4-common-effect-eliminate.test
***      Executed FactorEqualityTest
*** PASS: test_cases\q4\5-grade-var-elim.test
***      Executed FactorEqualityTest
*** PASS: test_cases\q4\6-large-bayesNet-elim.test
***      Executed FactorEqualityTest

```

Question q4: 2/2

Question q5

=====

```

*** PASS: test_cases\q5\1-DiscreteDist.test
***      PASS
*** PASS: test_cases\q5\1-DiscreteDist-a1.test
***      PASS
*** PASS: test_cases\q5\1-ObsProb.test
***      PASS

```

Question q5: 1/1

Question q6

=====

```

*** q6) Exact inference stationary pacman observe test: 0 inference errors.
*** PASS: test_cases\q6\1-ExactUpdate.test
*** q6) Exact inference stationary pacman observe test: 0 inference errors.
*** PASS: test_cases\q6\2-ExactUpdate.test
*** q6) Exact inference stationary pacman observe test: 0 inference errors.
*** PASS: test_cases\q6\3-ExactUpdate.test
*** q6) Exact inference stationary pacman observe test: 0 inference errors.
*** PASS: test_cases\q6\4-ExactUpdate.test

```

Question q6: 2/2

Question q7

=====

```

*** q7) Exact inference elapsedTime test: 0 inference errors.
*** PASS: test_cases\q7\1-ExactPredict.test
*** q7) Exact inference elapsedTime test: 0 inference errors.
*** PASS: test_cases\q7\2-ExactPredict.test
*** q7) Exact inference elapsedTime test: 0 inference errors.
*** PASS: test_cases\q7\3-ExactPredict.test
*** q7) Exact inference elapsedTime test: 0 inference errors.
*** PASS: test_cases\q7\4-ExactPredict.test

```

Question q7: 2/2

Question q8

=====

```

*** q8) Exact inference full test: 0 inference errors.
*** PASS: test_cases\q8\1-ExactFull.test
*** q8) Exact inference full test: 0 inference errors.

```

```

*** PASS: test_cases\q8\2-ExactFull.test
ExactInference
[Distancer]: Switching to maze distances
Average Score: 763.3
Scores:          778, 769, 759, 761, 776, 761, 758, 753, 763, 755
Win Rate:        10/10 (1.00)
Record:          Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
*** Won 10 out of 10 games. Average score: 763.300000 ***
*** smallHunt) Games won on q8 with score above 700: 10/10
*** PASS: test_cases\q8\3-gameScoreTest.test

```

Question q8: 1/1

Question q9

=====

```

*** q9) Particle filter initialization test: 0 inference errors.
*** PASS: test_cases\q9\1-ParticleInit.test
*** q9) numParticles initialization test: 0 inference errors.
*** PASS: test_cases\q9\2-ParticleInit.test

```

Question q9: 1/1

Question q10

=====

```

*** q10) Particle filter observe test: 0 inference errors.
*** PASS: test_cases\q10\1-ParticleUpdate.test
*** q10) Particle filter observe test: 0 inference errors.
*** PASS: test_cases\q10\2-ParticleUpdate.test
*** q10) Particle filter observe test: 0 inference errors.
*** PASS: test_cases\q10\3-ParticleUpdate.test
*** q10) Particle filter observe test: 0 inference errors.
*** PASS: test_cases\q10\4-ParticleUpdate.test
*** q10) successfully handled all weights = 0
*** PASS: test_cases\q10\5-ParticleUpdate.test
ParticleFilter
[Distancer]: Switching to maze distances
Average Score: 180.2
Scores:          188, 192, 198, 186, 167, 180, 184, 187, 164, 156
Win Rate:        10/10 (1.00)
Record:          Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
*** Won 10 out of 10 games. Average score: 180.200000 ***
*** oneHunt) Games won on q10 with score above 100: 10/10
*** PASS: test_cases\q10\6-ParticleUpdate.test

```

Question q10: 2/2

Question q11

=====

```

*** q11) Particle filter full test: 0 inference errors.
*** PASS: test_cases\q11\1-ParticlePredict.test
*** q11) Particle filter full test: 0 inference errors.
*** PASS: test_cases\q11\2-ParticlePredict.test
*** q11) Particle filter full test: 0 inference errors.
*** PASS: test_cases\q11\3-ParticlePredict.test
*** q11) Particle filter full test: 0 inference errors.

```

```

*** PASS: test_cases\q11\4-ParticlePredict.test
*** q11) Particle filter full test: 0 inference errors.
*** PASS: test_cases\q11\5-ParticlePredict.test
ParticleFilter
[Distancer]: Switching to maze distances
Average Score: 382.8
Scores:          386, 389, 363, 388, 388
Win Rate:        5/5 (1.00)
Record:          Win, Win, Win, Win, Win
*** Won 5 out of 5 games. Average score: 382.800000 ***
*** smallHunt) Games won on q11 with score above 300: 5/5
*** PASS: test_cases\q11\6-ParticlePredict.test

```

Question q11: 2/2

Finished at 16:54:42

Provisional grades

=====

Question q1: 2/2
 Question q2: 3/3
 Question q3: 2/2
 Question q4: 2/2
 Question q5: 1/1
 Question q6: 2/2
 Question q7: 2/2
 Question q8: 1/1
 Question q9: 1/1
 Question q10: 2/2
 Question q11: 2/2

Total: 20/20

Your grades are NOT yet registered. To register your grades, make sure to follow your instructor's guidelines to receive credit on your project.

Q1.

1. 定義 variable 跟 edge

- variables: 將 PAC, GHOST0, GHOST1, OBS0, OBS1 都加入變數中
- edges: 將每個 ghost 跟 pac 跟 observation 的關係建立成 edge，如此可反映出 observation 的值同時與 pac 及 ghost 的位置相關

2. 設定 variableDomainsDict:

- 透過 wall (X_RANGE, Y_RANGE) 取得所有可移動的 positions
- 將 PAC, GHOST0, GHOST1 的可能移動位置放入 variableDomainsDict
- 計算 observation 的可能範圍
 - 使用 Manhattan distance 計算所有位置的最大距離
 - 根據 max possible distance 加上 MAX_NOISE 噪點
 - 將 OBS0, OBS1 的可能範圍放入 variableDomainsDict

Q2.

1. 初始化 unconditioned_vars 及 conditioned_vars

2. 針對所有的 factor 進行資訊蒐集，將 factor 的 uncondition 及 condition 資訊放入上述的變數中
3. 假設 Bayes 中的每一個 factor variable domains dict 相同，取第一筆作為 new factor 的 variable domain
4. 取得所有可能的 assignment，並將所有的 factor 的機率去做乘積計算 probability，並且賦值給 new factor
5. 回傳 new factor

Q3.

1. 建立 new_unconditioned_vars, new_conditioned_vars 及 new_factor，並初始化每一個 new_factor 的 probability 為 0 (根據 assignments)
2. 根據所有可能的 assignment 計算 eliminate 後的機率，並賦值給 new factor。
 - a. 取得未被 eliminate 的值
 - b. 計算 old_assignment 及 new_assignment 的機率並且加總後取得新的機率和。用意是希望能夠結合新舊的 assignment (包含被 eliminate 等資訊) 都能被保留下來不會遺失
3. 回傳 new factor

Q4.

1. 透過 bayesNet.getAllCPTsWithEvidence 進行 factors 的初始化，取得所有的 factor
2. 針對每個 eliminationOrder 處理每個 variable
 - a. 透過 joinFactorsByVariable 合併 factors
 - b. 根據條件判斷是否有 unconditionedVariables，若有則進行 elimination，否則不進行處理
3. 透過 joinFactors 合併所有的 factor
4. 回傳 normalized 後的結果

Q5a.

1. normalize:
 - a. 先計算所有在這個 distribution 的值總合
 - b. 針對每個 distribution 的 key 值。將每個值與總和相除，確保分布於 0 ~ 1 且加總為 1，已建立全部的 probability
2. sample:
 - a. 計算 cumulative probability
 - b. 透過 random.uniform 產生 0 ~ cumulative prob 所加總的 total sum
 - c. 因為 cumulative prob 是累加的，所以可基於 key 遞迴處理 cumulative probability，當 random 小於 key 的機率時，回傳 key 值

Q5b.

1. 透過 util.manhattanDistance 計算 pacman 跟 ghost 的真實距離
2. 計算在給定真實距離的時候，observation 到 noise distance 的機率，條件如下 pseudo code:

```
if ghostPosition == jailPosition:
    return 1.0 if noisyDistance is None else 0.0
elif noisyDistance is None:
    return 0.0
else:
    return busters.getObservationProbability(noisyDistance, trueDistance)
```

Q6.

1. 透過 self.getJailPosition 取得 jail 的位置
2. 初始化 pacman position

3. 初始化 new beliefs：透過 DiscreteDistribution 建立一個新的 distributioin，儲存相關 distribution
4. 根據所有 position (表示 ghost 可能的位置) 進行處理
 - a. 透過 self.getObservationProb() 函數計算給定 Pacman 位置、幽靈位置、監獄位置和觀測值的條件下，觀測到當前觀測值的機率 (likelihood)
5. 將 beliefs 更新為 newBeliefs 並且進行 normalization

Q7.

1. 初始化 new beliefs，取得新的 beliefs distribution
2. 針對 all position 取得所有 ghost 的目前可能位置，並且計算其轉移到新位置的機率分布，亦即從 old position 透過 position distribution 轉移到 new pos distribution 的機率
3. 更新 newBeliefs 的新位置的 belief 機率
4. 將 beliefs 更新為 newBeliefs 並且進行 normalization

Q8.

1. 初始化最小距離
2. 找到最近的 GHOST 距離，更新 minDist 及 GHOST position
3. 從所有的合法行為當中，計算 pacman 的下一個位置及跟 ghost 的距離，若距離小於 minDist，則更新最佳行為及最短路徑

Q9.

- initializeUniformly:
 1. 將 particles 的總數除以合法位置的數量，以確定每個位置應該放置多少 particles
 2. 如果不能完全均勻地分配，將多餘的粒子均勻地分配到所有位置
 3. 將這些分配好的粒子放入 self.particles 列表中
- getBeliefDistribution:
 1. 創建一個空的 discrete distribution
 2. 針對每個 particle 將其對應的位置 belief += 1
 3. 對 belief_distribution 進行 normalization

Q10.

1. 取得幽靈的監獄位置和 Pacman 的位置
2. 建立 discrete distribution new_belief 來儲存更新後的 belief distribution
3. 處理每個 particles：
 - 使用 getObservationProb 計算觀察值對應的概率，即 GHOST 在當前 particles position 的 probability
 - 將計算得到的 probability 添加到 new belief 對應 particle position
4. 檢查新的 new belief distribution 的總和是否為零。如果為零，表示所有 particles 的權重都為零，需要透過 initializeUniformly 重新初始化
5. 如果總和不為零，則將新的 new belief distribution 進行 normalization，並根據新的 distribution 重新對 particles 進行 sampling，更新 self.particles

Q11.

1. 創建一個新的空列表 new_particles 儲存更新後的粒子狀態。
2. 針對每個 particles 進行處理：
 - 使用 getPositionDistribution 取得當前 particles 在遊戲狀態下的位置分佈
 - 從該位置分佈中取樣一個新的位置作為粒子的下一個狀態
 - 將新的 particle 狀態添加到 new_particles 中。

3. 將 new_particles 更新給 self.particles