REPORT

Programming Questions

Reinforcement learning

Question1 PFA ValueIterationAgents.py

Question 2

Values tried but failed:

- Discount =0.5

- noise = 0.4

Value for which got success

- Noise = 0.01

- Noise =0

Reason why it worked:

It worked for lower values of noise(0 and .01), the reason behind it is that with less noise agent has **more flexibility to learn path** as compared with high noise learning because noise prevents agent from taking desired action and if that value is high, convergence of MDP is difficult.

Question 3)

1)

Failed for :

- discount = 0.9 noise =0 reward =0.8

- discount = 1 noise =0 reward =0.8

worked for discount= 0.1 noise =0 reward = (0.1 – 0.8)

Reason for Working:

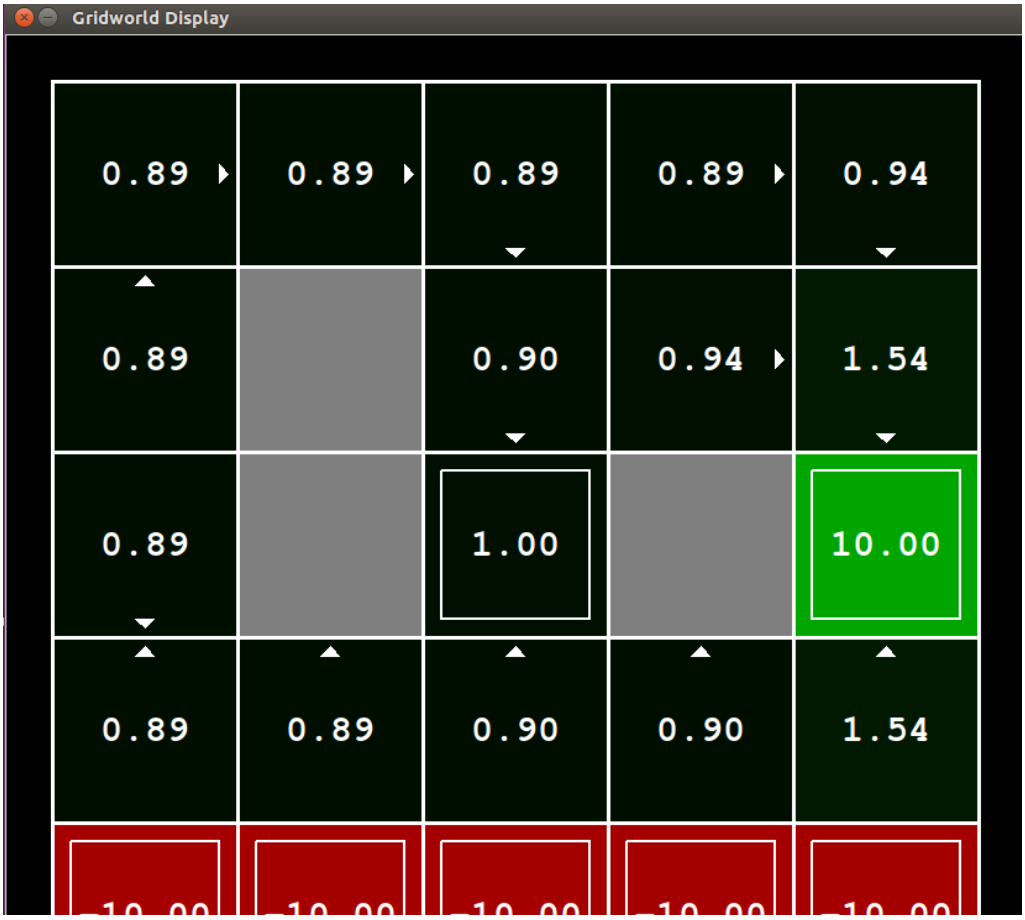
It is working for discount = 0.1 because if the discount is less, then it looks for nearest terminal state. So, it is going to 1 instead of 10. Also noise is kept as 0 which is making transition probability as 1. It is working for reward value from 0.1 to 0.8. Higher the reward, higher will be the return.

2)

Failed:

- discount= 0.1 noise =0.2 reward = 0.8

- discount= 0.1 noise =0.3 reward = 0.8



worked for discount= 0.1 noise =0.1 reward = (0.1 – 0.8)

Reason for Working:

It is working for discount = 0.1 because if the discount is less, then it looks for nearest terminal state. So, it is going to 1 instead of 10. Also noise is kept as 0.1 which is causing agent to take longer path. It is working for reward value from 0.1 to 0.8. Higher the reward, higher will be the return.

3)

Failed for

- discount= 1 noise =0 reward = 0.9

- discount= 1 noise =0.5   reward = 0.9

worked for discount= 0.9 noise =0 reward = (0.1 – 0.9)

Reason for Working:

It is working for discount = 0.9 because if the discount is more, then it looks for farthest terminal state. So, it is going to 10 instead of 1. Also noise is kept as 0 which is making transition probability 1. It is working for reward value from 0.1 to 0.9. Higher the reward, higher will be the return.

4)

Failed for:

- discount= 0.9 noise =0 reward = 0.9

- discount= 0.9 noise =1 reward = 0.9

worked for discount= 0.9 noise =0.2 reward = (0.1 – 0.9)

Reason for Working:

It is working for discount = 0.9 because if the discount is more, then it looks for farthest terminal state. So, it is going to 10 instead of 1. Also noise is kept as 0.2 which is causing agent to take longer path. It is working for reward value from 0.1 to 0.9. Higher the reward, higher will be the return.

5)

Failed for :

- discount= .5 noise =0.7   reward = 0.5

- discount= .6 noise =0.7 reward = 0.5

worked for discount= 0 noise =0 reward = 0.5

Reason for Working:

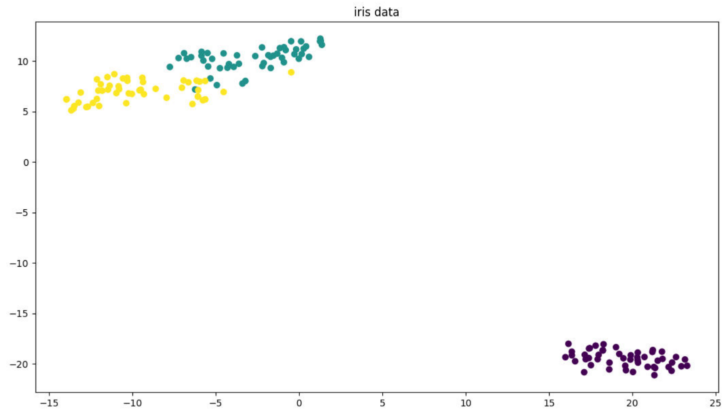
It is working for discount = 0 because if the discount is 0 , then iot will stop after first iteration, changing reward has no effect in the decision.

K-Means

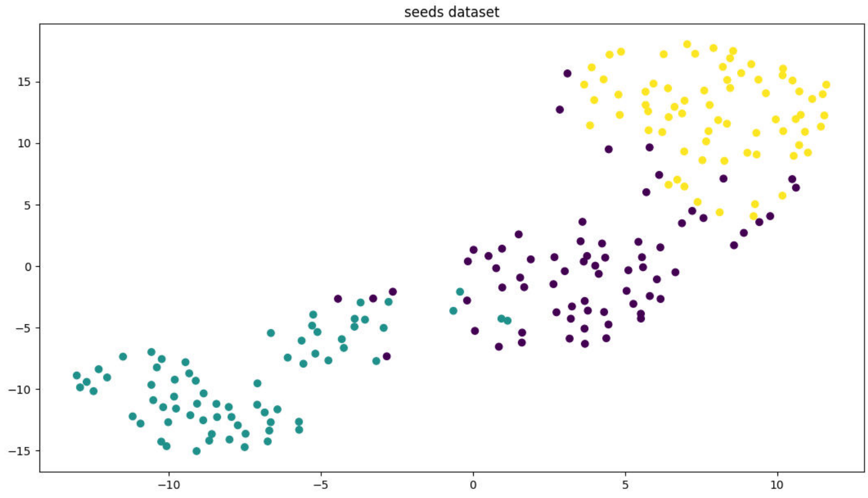
Qualitative Analysis

1.

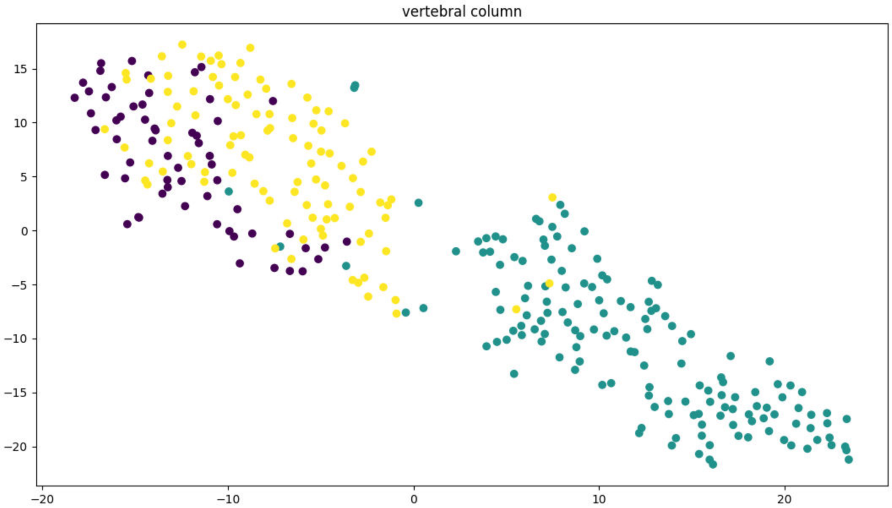
- Iris Dataset



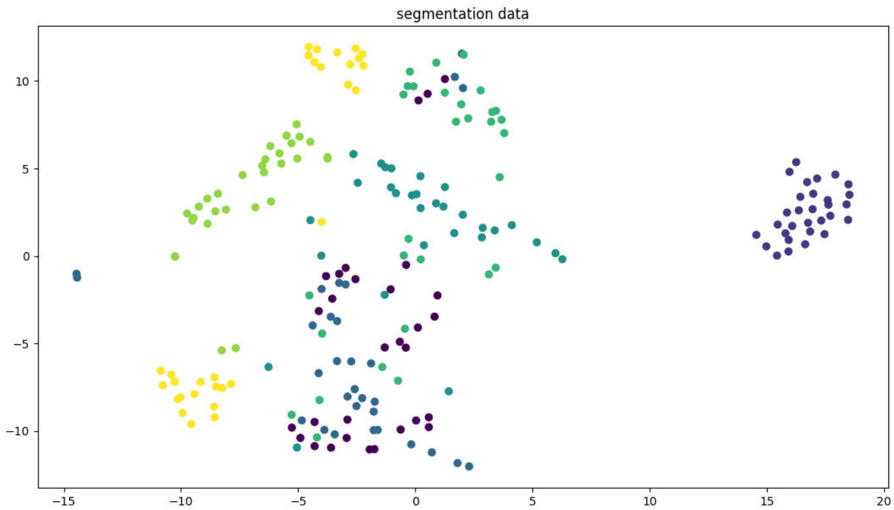
- Seeds dataset



- Vertebral dataset

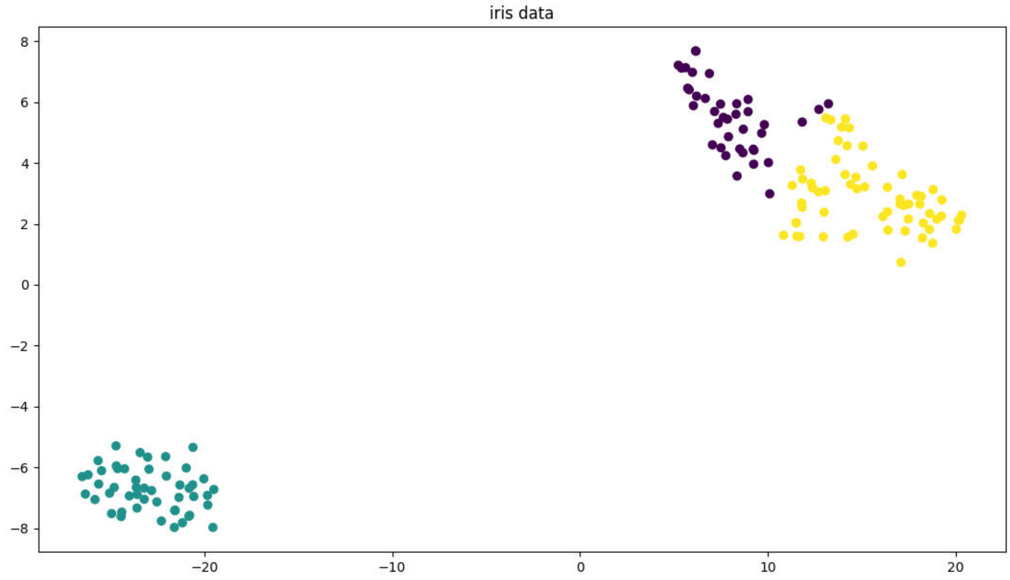


- Segmentation Data

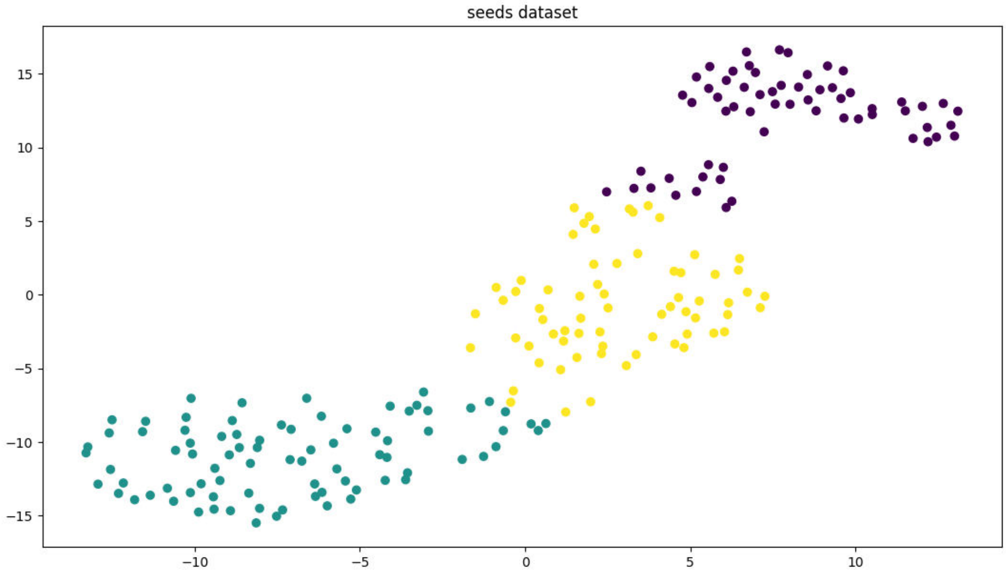


2.

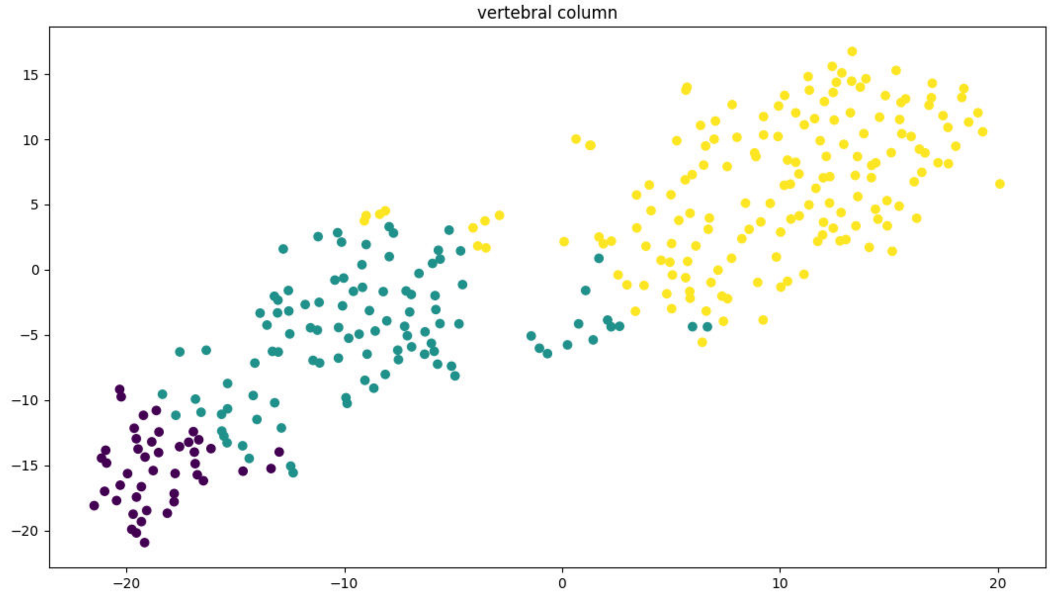
- Iris data



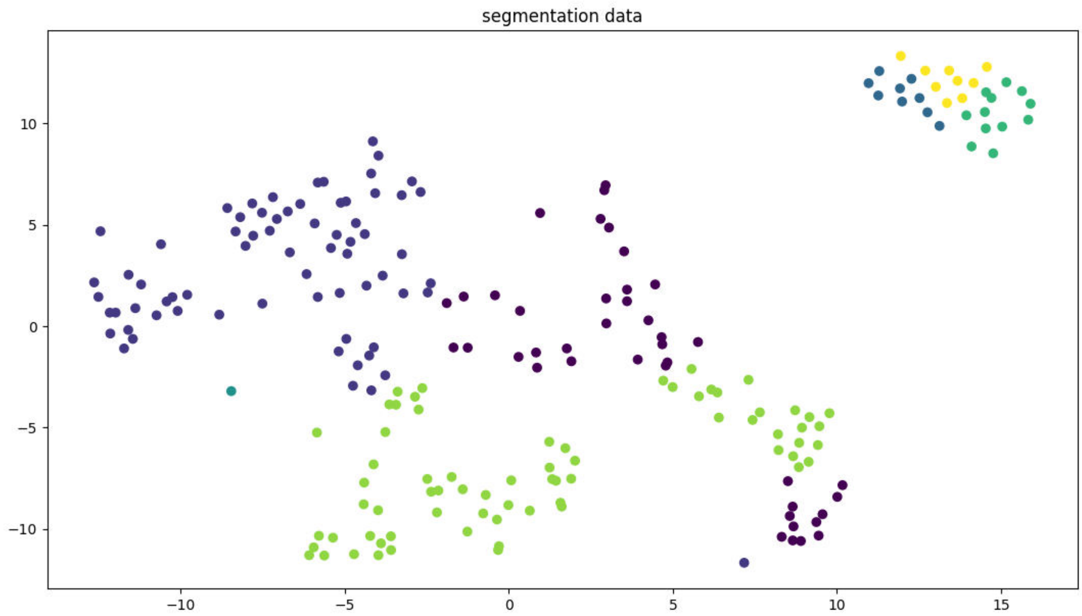
- seeds Data



- Vertebral data

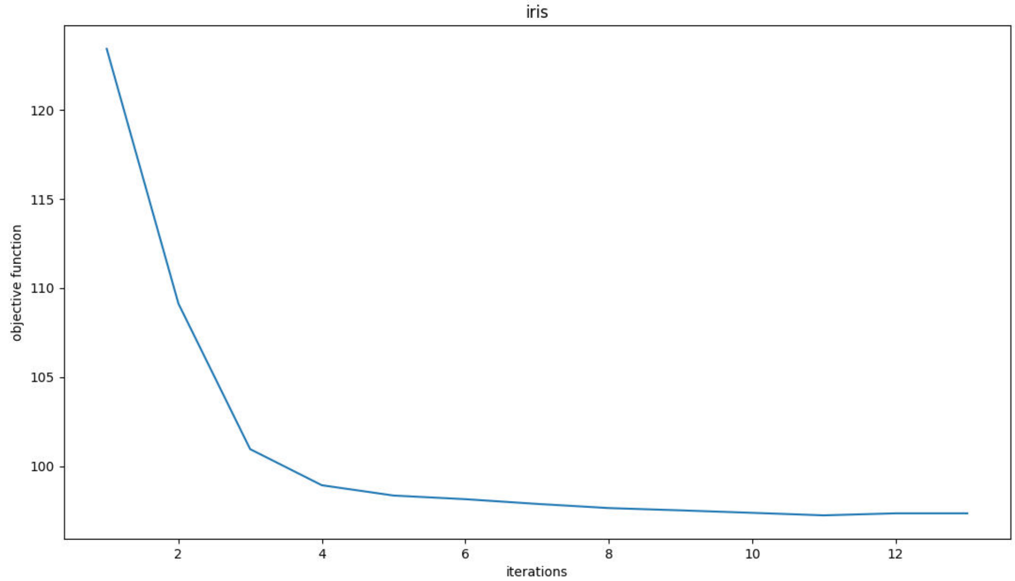


- segmentation Data

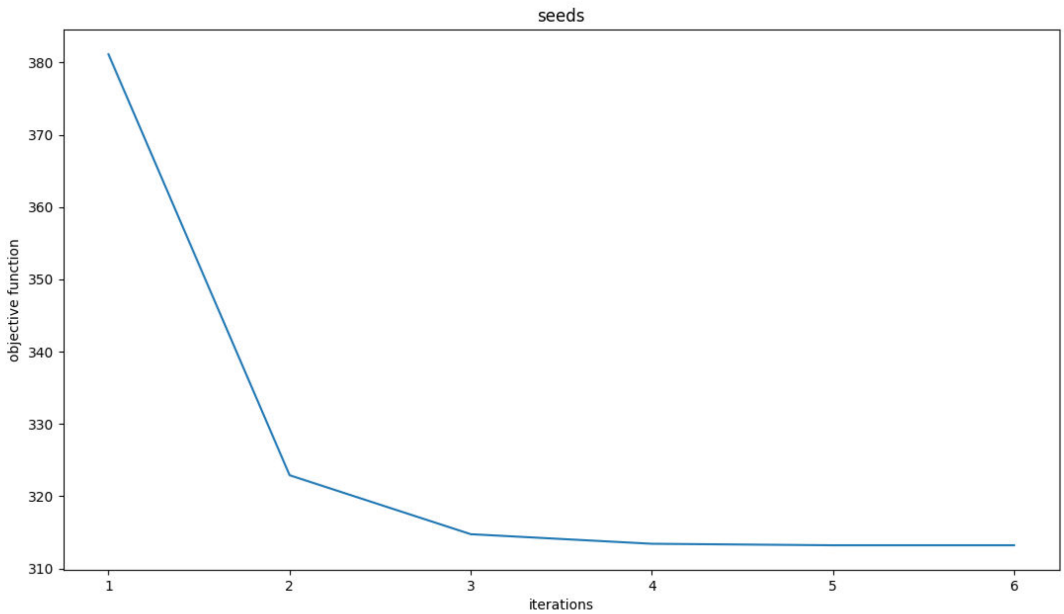


3.

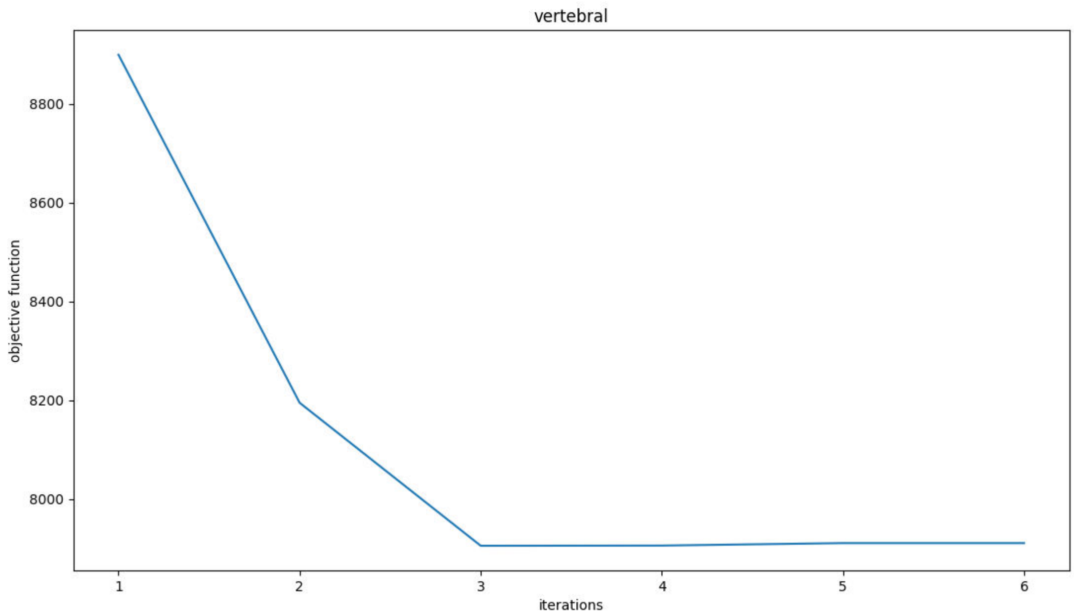
- iris data



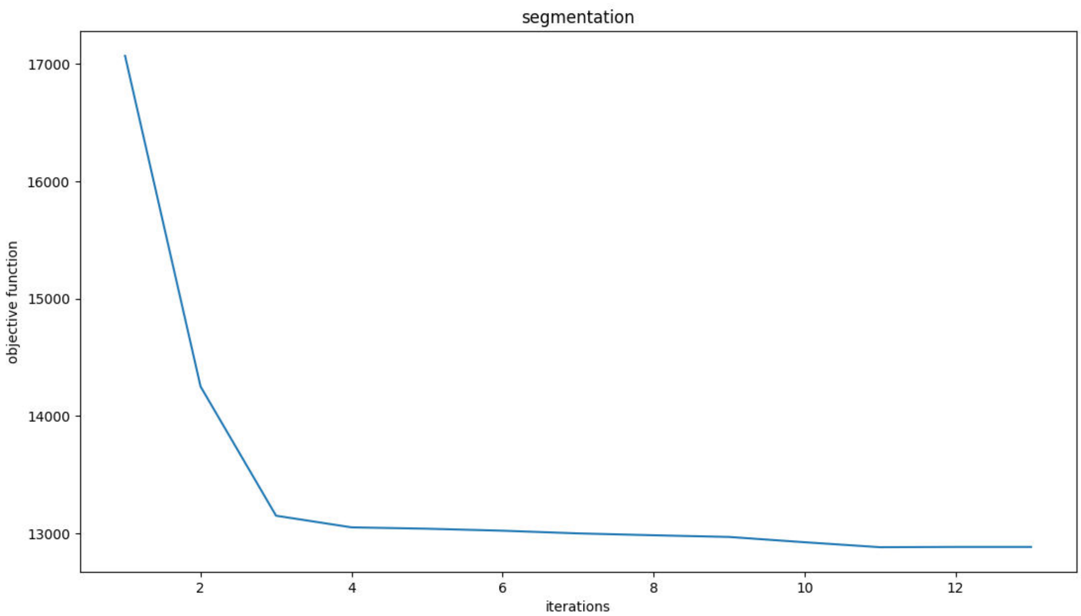
- Seeds data



- Vertebral data



- Segmentation data



Quantitative Analysis:

Data	k=2	k=2	k=2	k=true	k=true	k=true	k=12	k=12	k=12
	ARI	NMI	AMI	ARI	NMI	AMI	ARI	NMI	AMI
IRIS	0.5399	0.6793	0.5194	0.7219	0.7484	0.7392	0.3312	0.6177	0.4026
SEGMENTATION	0.0797	0.3272	0.1497	0.3737	0.5316	0.4841	0.4135	0.6053	0.5205
SEEDS	0.4664	0.5512	0.4259	0.715364	0.6979	0.6936	0.2776	0.5345	0.3475
VERTEBRAL	0.2972	0.4239	0.3340	0.3075	0.4153	0.4059	0.1782	0.4030	0.2579

Based on Qualitative analysis:

- Clustering of seeds data and iris data is good.this is evident from graphs in part 2.
- Objective function/cost is also less for seeds and iris data which implies that points in a cluster are more closed to its centroid.
- Clustering of verterbral data and segmentation data is not good, this is due to the reason that scattering/ outliers are more in them.
- Cost for verterbral and segmentation is also high(evident in part 3). this implies that distance of points in cluster with centroid is large.

Based on Quantitative analysis:

- Values of ARI,NMI and AMI is good for iris and seeds dataset for true values of cluster.This implies that clustering is good.
- Values of ARI,NMI and AMI is bad for segmentation and verterbral dataset for true values of cluster.This implies that clustering is not good.
- For lower value of k in segmentation data, values are worse as we are trying to cluster 7 different classes into 2.

Consistency:

- Both qualitative and quantitative estimates are depicting same results.
- Quantative assessment is better because it is comparing true labels with actual ones which is not the case for Qualitative one

Theory Questions

1. a)

DATE
PAGE

Assumption: Reward is taken of state S

(a) $V_1(S_0) = \left[\{0.5 \times (\cancel{0.9}\ 1 + 0.9 \times 0)\} + \{0.5 \times (1 + 0.9 \times 0)\}\right]$

$= [0.5 + 0.5]$

$= 1$

$V_1(S_1) = \max\left[\{[1 * (2 + 0.9 \times 0)]\},\ \{[0.5 \times (2 + 0.9 \times 0) + 0.5 \times (2 + 0.9 \times 0)\}\right]$

$= \max[\{2, 2\}]$

$= 2$

$V_1(S_2) = 1 \times [3 + 0.9 \times 0]$

$= 3$

$V_1(S_3) = 1 \times [10 + 0.9 \times 0]$

$= 10$

after iteration 1:

$V_1(S_0) = 1 \quad V_1(S_1) = 2 \quad V_1(S_2) = 3 \quad V_1(S_3) = 10$

Iteration 2:

$V_2(S_0) = \left[(0.5 \times (1 + 0.9 \times 2)) + (0.5 \times (1 + 0.9 \times 3))\right]$

$= [0.5 \times 2.8 + 0.5 \times 3.7]$

$= 3.25$

$V_2(S_1) = \max\left[(1 \times (2 + 0.9 \times 3)),\ (0.5 \times (2 + 0.9 \times 10) + 0.5 \times (2 + 0.9 \times 2))\right]$

DATE
PAGE

$= \max[4.7,\ 5.5+1.9]$
$= \max(4.7,\ 7.4)$
$= 7.4$

$V_2(S_2) = 1 \times (3 + 0.9 \times 1)$
$= 3.9$

$V_2(S_3) = 1 \times (10 + 0.9 \times 10)$
$= 19$

$\boxed{\text{after iteration 2:} \quad V_2(S_0) = 3.25 \quad V_2(S_1) = 7.4 \quad V_2(S_2) = 3.9 \quad V_2(S_3) = 19}$

Iteration 3:

$V_3(S_0) = [0.5 \times (1 + 0.9 \times 7.4) + 0.5 \times (1 + 0.9 \times 3.$

$= 3.83 + 2.255$
$= 6.085$

$V_3(S_1) = \{ \max[(1 * 2 * (0.9 \times 3.9)),$
$(0.5 \times (2 + 0.9 \times 19) +$
$0.5 \times (2 + 0.9 \times 7.4)]$

$= \max[5.51,\ 9.55 + 4.33]$

$= 13.88$

$V_3(S_2) = 1 \times (3 + 0.9 \times 3.25)$
5.925

$V_3(S_3) = 1 \times (10 + 0.9 \times 19)$

27.1

after iteration 3.

$V_3(S_0) = 6.085$
$V_3(S_1) = 13.88$
$V_3(S_2) = 5.925$
$V_3(S_3) = 27.1$

1 b)

(b) Optimal Policy at state S_1

$$\pi[S_1] = \text{argmax}_a \sum_{S'} P(S'|S_1, a) \left(R(S_1, a, S') + \gamma V_K[S']\right)$$

From S_1, there are two actions, let us called them as a_{11} & a_{12}.

For action a_{11},

Possible state $S' = S_2$

$$\therefore Q = P(S_2 | S_1, a_{11}) \times \left[R(S_1, a_{11}, S_2) + \gamma \left(V_k[S_2] \right) \right]$$

Calculating value for $k = 3$

$$1 \times [2 + 0.9 \times 5.925]$$

$$= 7.3325$$

For action a_{12}

Possible states $\Rightarrow S_1$ & S_3

$$\therefore Q = P(S_1 | S_1, a_{12}) \times \left[R(S_1, a_{12}, S_1) + \gamma V_3[S_1] \right]$$

$$+ P(S_3 | S_1, a_{12}) \times \left[R(S_1, a, S_3) + \gamma V_3[S_3] \right]$$

$$= 0.5 \times [2 + 13.88 \times 0.9] +$$

$$0.5 \times [2 + 0.9 \times 27.1]$$

$$= 7.246 + 13.195$$

$$= 20.441$$

(b) Optimal Policy at state S_1

$$\pi[S_1] = \arg\max_a \sum_{s'} P(s' | S_1, a) \left(R(S_1, a, s') + \gamma V_K[s'] \right)$$

1. c) True or False

i. False

ii. False

iii. True, if the discount factor is 0, there will be no change in value for subsequent iterations as the term (discount* value from previous iteration) will not be changed. So, MDP wil converge in 1 iteration only.

iv. True, since there are no cycle, after every iteration atleast one state will achieve its optimal value in each iteration and this will go on it all states are converged.

v. False

2. Note:Taking assumption that each color (R,G,B) requires 8 bits to represent intensity values from 0 to 255. So, total 24 bits are required to represent a pixel.

Number of bits required to transmit the image = (N * N *24) bits

After k-means clustering with k=K, labels for each pixel and centroids of cluster is to be stored, so total number of bits = $K*24 + N*N*\log_2 K$

Compression ratio = $(N*N*24) / (K*24 + N*N*\log_2 K)$